2

Naval Research Laboratory

Washington, DC 20375-5000

NRL Memorandum Report 6229

CARYSPEC

A FORTRAN 77 Program for Spectral Data Acquisition and Control of The Varian CARY 2390 UV-VIS-NIR Spectrophotometer

JOHN C. COOPER

Polymeric Materials Branch
Chemistry Division

ROBERT A. BINSTEAD

GEO-CENTERS, Inc.
10903 Indian Head Highway
Fort Washington, MD 20744

July 14, 1988

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
NRL Memorandum Report 6229

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Research Laboratory

6b. OFFICE SYMBOL *(If applicable)*
Code 6125

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Washington, DC 20375-5000

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
NAVSEA and ONR

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
Washington, DC 20362-5101 (NAVSEA)
800 N. Quincy St. Arlington, VA 22217 (ONR)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
65803N 61153N	RR011- 11-41	65803N 61153N	

11. TITLE *(Include Security Classification)*
CARYSPEC - A FORTRAN 77 Program for Spectral Data Acquisition and Control of the Varian CARY 2390 UV-VIS-NIR Spectrophotometer

12. PERSONAL AUTHOR(S)
Binstead,* R.A. and Cooper, J.C.

13a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
1988 July 14

15. PAGE COUNT
82

16. SUPPLEMENTARY NOTATION
*GEO-CENTERS, Inc., 10903 Indian Head Highway, Fort Washington, MD 20744

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
UV-VIS-NIR spectrophotometer, Data acquisition
HP1000 Interfacing

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

CARYSPEC is a FORTRAN 77 program designed for acquisition of UV-VIS-NIR spectra from the CARY 2300 - 2400 series spectrophotometers via an IEEE-488 interface bus to an external computer system. The program is written to operate on a Hewlett-Packard 1000 minicomputer but with very few system dependent features to enable easy conversion for other host systems. The operation of the spectrophotometer is controlled by CARYSPEC using menu displays on the system console to setup instrument parameters, baseline correction, data acquisition and disk file storage. CARYSPEC provides detailed error trapping for inappropriate instrument settings and automatic adjustment of spectral bandwidth and gain level during data acquisition to match the current baseline correction. Spectra are stored in disk files for subsequent use by plotting, editing and data analysis programs.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
John C. Cooper

22b. TELEPHONE *(Include Area Code)*
(202) 767-3115

22c. OFFICE SYMBOL
Code 6125

CONTENTS

Section	Description	Page
INTRODUCTION		1
IMPLEMENTATION		
1.0	Hardware Interface	2
1.1	System Handshaking	2
1.2	Instrument Commands Format	3
1.3	Instrument Commands Summary	7
1.4	Real Time Transmission Data Format	9
1.5	Illegal Parameter Changes	10
SOFTWARE DESCRIPTION		
2.0	Purpose Of CARYSPEC	11
2.1	Language Features Of CARYSPEC	11
2.2	Structure Of CARYSPEC	12
2.3	BLOCK DATA	22
2.4	SUBROUTINES	24
2.5	FUNCTIONS	31

✓

A-1

... cont'd

Section	Description	Page
PROGRAM CODE		
3.0	Source Code Availability	31
3.1	Variable Names And Usage	31
3.2	Program Listing	36

LIST OF TABLES

TABLE	Description	Page
I	ASCII Codes For Touch Panel Keys	6
II	Data File Format	21
III	Glossary Of Integer Variables	32
IV	Glossary Of Real Variables	33
V	Glossary Of Character Variables	34

CARYSPEC
A FORTRAN 77 Program For Spectral Data Acquisition And Control Of The Varian CARY 2390 UV-VIS-NIR Spectrophotometer

INTRODUCTION

The CARY 2300 and 2400 series spectrophotometers are high quality, microprocessor controlled analytical instruments intended for measurements of the UV-Visible and Near IR absorption spectra of solids, liquids and gases. When equipped with an optional IEEE-488 standard interface these instruments and their accessories are programmable by an external computer system enabling acquisition of spectral measurements in digital form. This document describes a fully tested FORTRAN 77 program, CARYSPEC, designed for *single scan* acquisition of spectra from a CARY 2390 instrument using a Hewlett-Packard 1000 minicomputer running the multi-user CI shell and RTE-6/VM operating system. The program uses very few machine specific functions and could be modified easily to run on other host systems supporting the IEEE-488 interface standard.

CARYSPEC implements a large subset of the programmable instrument control functions of the CARY 2390 spectrophotometer in a menu driven format closely resembling the menu displays on the instrument. Therefore, no special training is required for users already familiar with operation of the instrument. Indeed, operation from a computer console has proven to be more convenient than using the clumsy keypad on the instrument. The control functions implemented reproduce the facilities of the spectrophotometer's Instrument Settings, Baseline Setup, Lamp and Detector Modes and Accessory Setup menus, as well as a number of single keypad functions. The program *does not support* the statistics calculation modes, rapid scan setup keys or the automatic sequencer operation for repetitive scans. The latter mode, although useful, is not compatible with reliable IEEE-488 data transmission. A future version of the program may implement this feature using the the single scan mode of CARYSPEC with automatic data file storage and system clock control of the timing of successive scans.

CARYSPEC is a moderately large program and will not run in a single 32K word segment of the Hewlett-Packard 1000 minicomputer. The program has been segmented to run in 5 memory resident nodes of the HP 1000 using the Multi Level Segmentation utility programs SGMTR and MLLDR. The program requires an 83K word memory partition, including 40K words of Extended Memory Addressing (EMA) space for data arrays.

Manuscript approved February 29, 1988.

IMPLEMENTATION

1.0 Hardware Interface:

The IEEE-488 interface for the HP 1000 system is implemented with an HP 59310B interface card which utilizes 4 Logical Unit (LU) addresses in the system. The LU addresses are dependent on the computer system and are defined by the system generation. The interface card used by the program, CARYSPEC, occupies LU addresses 35-38 corresponding to card addresses 0-3. Address 0 is a special addressing mode which allows access to low level IEEE-488 bus command sequences for any device number. Addresses 1-3 are predefined automatic READ/WRITE channels which select device numbers 1-3 on the bus. CARYSPEC uses automatic device addressing from channel 3 (LU 38) and consequently the device address of the CARY 2390 has been set to 3. Physical connection between the computer and spectrophotometer consists of 5 metres of IEEE-488 standard cabling. Reliable data transmission was obtained with cable lengths up to 9 metres. With 13 metres of cable commands sent to the CARY 2390 were accepted but data transmissions from the CARY 2390 spectrophotometer were corrupted owing to the limited drive capability of the MC3447L bus transceivers in the CARY 2390. The bus was found to be totally inoperative with a cable length of 17 metres.

1.1 System Handshaking:

The IEEE-488 subsystem of the HP 1000 is operated by the RTE driver DVA37 configured for ASCII Data Record mode. This mode sends and expects to receive an End Of Record (EOR) with data transmission in the form of a Carriage Return/Line Feed (CR/LF) sequence, though Line Feed alone is sufficient. The CARY 2390 accepts commands in this format automatically. However, the instrument must be instructed to send data with the CR/LF trailer using the command "@@20", sent as a series of ASCII characters to the instrument. This is the first instrument command in the program CARYSPEC and instructs the CARY 2390 both to insert CR before LF on transmissions and to turn off check sum error detection. Therefore, all commands to the instrument end with a check sum byte of zero. Correctly operating IEEE-488 systems do not require check sum error detection. During real time spectral data transmission the CARY spectrophotometer is the active talker on the bus. The End Of Record character for this mode has been specified as a Line Feed in CARYSPEC - this works without a CR, as expected for IEEE-488 handshaking. Termination of real time transmission is effected by the active controller via a bus level routine which sends the ASCII codes for UNTALK/UNLISTEN.

1.2 Instrument Commands Format:

Programmable control of the CARY 2300–2400 series spectrophotometers has been implemented by Varian Instruments with a series of *single character* commands in ASCII code, most being accompanied by following characters to select a particular setting for the command selected. The instrument generates similar reply messages to most commands and these character strings must be read by the controller before sending further commands.

Varian's documentation defines the command structure as a sequence of [ASCII] characters in the following format (blanks added for clarity):

[LDI] [MI] [MD] [MQ] [CSM] [EOI]

where,

[LDI] = Logical Device Identifier

ASCII representation of the talk address of the sender – the value is ignored but some character must be sent as a place holder. {NOTE: The correct character for the bus controller at address 0 is "@"}.

[MI] = Message Identifier

ASCII character C, P or N used to indicate the message type as Command, Positive reply or Negative Reply. {NOTE: This character field is actually INVALID within a command – 'C' will cause a system reset if used}.

[MD] = Message Descriptor

ASCII character which specifies the actual command to be executed.

[MQ] = Message Qualifier

A string of characters used to set one or more variables or operating modes.

[CSM] = Check sum

The binary sum of all characters in a particular message - this is always the "0" character since checksum mode is turned off normally.

[EOI] = End Or Identify (actually End Of Record)

A linefeed character is specified as the terminator character to end data transmissions. {NOTE: This is appended automatically by most IEEE-488 drivers during handshaking.}

The [MI] field given in the message structure above is actually invalid within a command and must not be used - otherwise command 'C', system reset, will be executed followed by a bus hang up on the trailing unused command characters. However, the [MI] field is valid in the reply messages from the instrument.

The correct COMMAND format is given by the following fields:

[LDI] [MD] [MQ] [CSM] [LF]

EXAMPLE: Send the Record Trailer Set-up Command '@'

```
      Command='@@20'
      WRITE (UNIT=38,FMT=10) Command
 10   FORMAT (A4)
      READ (UNIT=38,FMT=20) Reply
 20   FORMAT (A64)
```

where,

@ = [LDI] address of bus controller
@ = [MD] command for record trailer set-up command
2 = [MQ] value to select no check sum (bit 1 = 0), insert CR before LF
0 = [CSM] check sum (off)
LF is sent automatically with WRITE command

All instrument commands sent to the CARY 2390 by an external computer comprise a sequence of ASCII data characters as far as the IEEE-488 bus is concerned. The details of the handshaking, with talk and listen addresses, are transparent to high level languages such as this implementation of FORTRAN 77 where such details are handled automatically by the device driver - in this case DVA37. Unfortunately, Varian Instruments chose to document the software control of the 2300 series instruments for a particular dialect of BASIC used in their proprietary controller, a model DS-15 data station, which appears to operate in a purely binary mode on the IEEE-488 bus. Thus, their examples of the message structure include a line feed character appended to the actual data command message. Furthermore, they confuse this End Of Record character (EOR) with the title of the End Or Identify handshake line of the IEEE-488 bus. The trailing line feed character is omitted from all instrument commands in the program CARYSPEC, this terminator being supplied automatically by the HP 1000 driver routine using the standard FORTRAN output command, WRITE.

While the reply messages generated by the CARY spectrophotometer must be read, only a few require testing for negative replies in a correctly structured program. CARYSPEC utilizes tight error trapping for inappropriate combinations of instrument parameters, diminishing the need for extensive use of the error message numbers from the instrument. In fact, only the Baseline Set Up subroutine checks for a negative reply and even that is probably superfluous since illegal combinations of operating modes are trapped before calling this routine. Such internal errror trapping provides a smoother user interface compared with taking corrective action after rejection of bad commands by the instrument.

The full range of instrument commands and their reply formats are summarized below in Section 1.3. Some commands can be accessed only using the 'D' command to mimic key pad presses on the instrument. These often involve sequences of key presses to implement a single function. Table I contains the ASCII codes required to send Key Pad entries with the 'D' command. However, full familiarity with the instrument is required to use these effectively. For example, the Baseline Set Up procedure could be implemented by sending a large number of Key Pad sequences but a more efficient means is the 'J' command which includes all of the requested baseline parameters in a single string.

TABLE I

ASCII Codes For Touch Panel Keys

KEY	DECIMAL	ASCII
0	48	'0'
1	49	'1'
2	50	'2'
3	51	'3'
4	52	'4'
5	53	'5'
6	54	'6'
7	55	'7'
8	56	'8'
9	57	'9'
.	58	':'
CLEAR	59	';'
CHANGE	60	'<'
ENTER	61	'='
ABS vs WLNGTH	64	'@'
ABS vs TIME	65	'A'
SEL WLNGTHS	66	'B'
INSTR SETTINGS	67	'C'
LAMPS & DETECTORS	68	'D'
AUTO OP	69	'E'
ACCRY SETTINGS	70	'F'
CALC MODE	71	'G'
BASLN SETUP	72	'H'
TEST FUNCTION	73	'I'
GOTO WLNGTH	74	'J'
LOCK	75	'K'
START	80	'P'
STOP	81	'Q'
RESUME	82	'R'
STANDBY	83	'S'
READY	84	'T'
AUTO BALANCE	85	'U'
CASSETTE	88	'X'
PRINT	89	'Y'
RIGHT CURSOR	104	'h'
LEFT CURSOR	105	'i'
MANUAL SCAN +	106	'j'
MANUAL SCAN -	107	'k'

1.3 Instrument Commands Summary:

'A' Lock or Unlock Keyboard

```
Command = '@AX0'                 where X = 0,1 (Unlock, Lock)
Reply   = '#PAX0'
```

'B' Status Request

```
Command = '@B0'
Reply   = '#PB[ data ]0'         5 bytes of data are returned
```

'C' System Reset

```
Command = '@C0'
No Reply
```

'D' Activate A Touch Panel Key

```
Command = '@DX0'                 where X = ASCII code for Key
Reply   = '#PDX0'
Reply   = '#NDX0'
```

'E' Dump Parameter Table

```
Command = '@E0'
Reply   = '#PE[ no. of data bytes ][ data ]0'
```

'F' Accessory ON/OFF Control

```
Command = '@FXY0'                where X = 0,1 (Turn Off, On)
                                 where Y = Accessory Number
Reply   = '#PFXY00'              where 0 before CSM = no error
Reply   = '#NFXY[ error no. ]0'
```

'G' Return Value Of Parameter Or Variable

```
Command = '@G1Y0'                Y = Index Number Of Parameter
Reply   = '#PG1Y[ string length ][ string ]0'
Command = '@G2Y0'                Y = Index Number Of Variable
Reply   = '#PG2Y[ value ]0'
Reply   = '#NGXY[ error no. ]0'   for X = 1,2
```

'H' Change Value Of A Parameter

```
Command - '@HXY0'                    where X = Parameter Number
                                     where Y - Required Index Value
Reply   - '#PHXYZ0'                  where Z - New Index Value
Reply   - '#NXY[ error no. ]0'
```

'I' Change Value Of A Variable

```
Command - '@IX[ number ]!0'          where X - Variable Number
Reply   - '#PIX0'
Reply   - '#NIX[ error no. ]0'
```

'J' Set Up A Baseline

```
Command - '@J[ value 1 ]!...[ value 9 ]!0'
Reply   - '#PJ00'                    where 0 before CSM - no error
Reply   - '#NJ[ error no. ]0'
```

'K' Request Real Time Data Transmission

```
Command - '@K1Y0'                    Interval mode
Command - '@K3Y[ interval ]!0'       Continuous mode
where Y - 0-3  specifies delimiter (0, LF, CR, &)
Reply   - '*[ data ]!...[ data ]![EOR]#...
Reply   - '#NKXY[ error no. ]0'      where X - 1,3 (mode)
```

'L' Display Message On Line 4 Of C.R.T.

```
Command - '@L1[ message ]0'
Command - '@L00'                     Turn Off Message Display
Reply   - '#PL0'                     No Negative Reply
```

'M' Accessory Mode Set Up

```
Command - '@MX[ value ]!0'           where X - Parameter Number
Reply   - '#PMX00'                   where 0 before CSM - no error
Reply   - '#NMX[ error no. ]0'
```

'@' Record Trailer Set Up

```
Command - '@@Y0'                     where Y - 0-3 (2 for CR/LF)
No reply
```

1.4 Real Time Transmission Data Format:

The 'K' command selects one of two real time transmission modes with the CARY spectrophotometer as the active talker on the bus. The continuous mode transmits data at the frequency of the instrument's chopper motor (15 Hz at line frequeny = 60 Hz) in an abbreviated format of Ordinate and Abscissa. The more useful mode, as used in CARYSPEC, is the programmed interval mode which transmits 9 instrument measurements. This increases the overhead for each datum but the extra string processing time has been found to be insignificant for the HP 1000 system. CARYSPEC limits the choice of scan speed and wavelength interval for a maximum transmission rate of 5 Hz. This modest rate is determined by the interrupt service times of the multi-user operating system rather than program processing speed.

The data format for the programmed interval mode varies with the choice of Ordinate and Abscissa modes for the CARY 2390. The data acquisition subroutine ACQUIRE within CARYSPEC supports all 6 choices of Ordinate mode and the 4 choices of Abscissa. However, the main portion of CARYSPEC rejects any choices other than Absorbance or Transmittance *vs* Wavelength which send data in the following formats:

A typical record for Absorbance *vs* Wavelength: (59 characters)

```
#   0.0012! 2000.00!1!01!128! 2000.00!    0.0! 28.72!-199.83!
```

A typical record for Transmittance *vs* Wavelength: (58 characters)

```
# 100.06! 2000.00!1!01!128! 2000.00!    0.0! 28.72!-199.73!
```

These fields correspond to Ordinate, Abscissa, Cell #, Cycle #, Sample #, Wavelength, Time, Temperature and Gel Scanner Distance. Transmissions from the CARY 2390 are read left-justified into a CHARACTER variable dimensioned to length 64. This is sufficient for all operating modes and makes ACQUIRE a general purpose subroutine for use in other programs. Since the record format is fixed for each choice of Ordinate and Abscissa there is no need to search the data strings for the '!' delimiters. CARYSPEC begins substring extraction at character position 2 and uses arrays XOFF(I) and YOFF(J) to determine the offsets for the first two data fields. The remaining substrings are fixed length and their boundaries are calculated from the sum of the lengths of the first two data fields.

1.5 Illegal Parameter Changes:

Several instrument parameters have been masked off from changes by an external computer, so becoming READ ONLY. The slit height parameters #22 & #26 are not programmable since the slit height is a manual adjustment. The Baseline parameters also are intended to be READ ONLY in order to prevent overwriting the descriptors for a current baseline. Hence, parameters #23 - #26 are updated only when a new Baseline request is sent using the 'J' command. The Baseline status parameter #37 has limited accessibility and can be turned ON or OFF only. CARYSPEC also allows parameter #37 to be set to the RECORD and ON/SETUP states by issuing Key Pad sequences with the 'D' command. However, CARYSPEC does not use these settings to actually record the Baseline. The settings are used only to transfer setup information between the instrument and baseline menu parameters for users accustomed to this feature.

Unfortunately, two setup parameters, DERIV TEMP RANGE (#11) and TEMP ZERO (#13), have also been masked off making it difficult to control the CARY in some operating modes. However, it was discovered that the DERIV TEMP RANGE can be set by using parameter #10, the DERIV RANGE settings for Absorbance and %T. Thus, parameter #11 appears to be an internal table only. Special action has to be taken in selecting the derivative range settings since only the 1,5,10 sequence is valid while a 1,2,5,10 sequence can be selected. CARYSPEC includes an INDEX array variable which holds the valid indices for the derivative modes. This allows derivative spectra to be drawn while the external computer acquires the raw measurements. While CARYSPEC does not allow acquisition with TEMPERATURE as the Ordinate or Abscissa, the functionality of the Temperature setup modes is preserved with one exception. The TEMP ZERO parameter can not be set from the external computer and only the range can be selected from CARYSPEC. Since this is not a feature required for CARYSPEC no attempt has been made to issue a Key Pad sequence for TEMP ZERO.

The CARY 2390 also masks off the %T offset variable (#10) when the 200 %T range is selected. This appears to be designed so that only a 0-200 %T range can be selected. However, if a non-zero offset has already been set for another scale then selection of the 200 %T range will not result in a 0-200 %T scale - it will have the old offset. This illegal mode can be reset by changing to another range and setting the offset to zero before selecting the 200 %T range again.

SOFTWARE DESCRIPTION

2.0 Purpose Of CARYSPEC:

The collection of spectrophotometric data in digitized form provides both a permanent means of storage and the ability to perform more sophisticated analysis. While the instrument obtains spectral measurements as absorbance *vs* wavelength (nm), plotting programs can rescale the raw data into more meaningful units such as molar absorptivity *vs* wavenumber (cm^{-1}). Techniques such as difference spectroscopy no longer need to be performed in real time since data files can be manipulated easily to achieve this function by scaling and subtraction. Noise can be removed from single scan spectra using least squares smoothing while similar functions can be used to generate derivative spectra which are more accurate than those produced in real time by the CARY 2300–2400 series spectrophotometers on their internal pen recorders. Such benefits make it worthwhile to develop software for data transfer between the CARY spectrophotometer and an external computer system, in this case a Hewlett–Packard 1000 minicomputer running the CI shell and RTE–6/VM operating system.

2.1 Language Features Of CARYSPEC:

The program CARYSPEC was written in FORTRAN 77 since this language provides the most complete set of interface and control functions available on the HP 1000. The communication between FORTRAN 77 and the IEEE–488 interface to the CARY 2390 spectrophotometer is completely transparent and standard READ/WRITE statements control the operation of the instrument and the collection of data transmitted by the CARY. Therefore, the program is portable, with some minor alterations, between systems supporting the FORTRAN 77 language and IEEE–488 Input/Output. CARYSPEC uses three machine specific function calls requiring substitution to run on a different host system. The first is CALL FFRCL(79) which changes the free field record length from the default value of 72 to 79. This is used to provide more column space on the console display screen. The second is a call to read the system clock to provide calibrated delay loops. Thus the operation of SUBROUTINE Wait(DELAY) and FUNCTION Time(I) would need to be altered. The third is CALL ABRT(35,3) which terminates transmission from the CARY by sending UNTALK/UNLISTEN on the IEEE–488 bus. Syntactical differences also appear between various versions of FORTRAN 77, particularly in the READ/WRITE statements. CARYSPEC uses the format READ (1,...) and WRITE (1,...) for the user's console (defined as LU 1) while Microsoft's compiler uses an * to denote the console unit.

2.2 Structure Of CARYSPEC:

CARYSPEC comprises a large main program unit containing most of the console menu displays, block data for named COMMON variables and a number of subroutines that perform string processing, input validation and communication with the CARY 2390 spectrophotometer. The main program is responsible for all the logic flow and the subroutines execute specific support tasks, which are summarized below:

The main program unit of CARY\PEC comprises 9 distinct segments of code to carry out the the functions of instrument setup, spectral data acquisition and disk file data storage. The code fragments appear under the following assigned labels: MENU, SPECTRUM, BASELINE, ADVANCED, INSTRUMENT, LAMP, ACCESSORY, STORE and EXIT.

MAIN PROGRAM

MENU:

This is the first and main control menu of the program, selecting entry to the remaining instrument control menus, data acquisition, storage and exit routines. The choices are as follows:

'A'.....Acquire Spectrum

This selection branches to label SPECTRUM and performs logical tests for the presence of a valid Baseline in the CARY, valid choices of Abscissa/Ordinate modes and the presence of an unstored spectrum in memory before proceeding with data acquisition. If there is no valid Baseline information in memory the program will branch to label BASELINE. If the Abscissa/Ordinate settings are inappropriate the program will branch to label INSTRUMENT.

'B'.....Baseline Setup

This selection branches to label BASELINE which reads the current instrument settings and presents the pertinent Baseline parameters in a menu arrangement similar to the equivalent display on the CARY. The user can alter these selections but most will not take effect unless a new Baseline scan is recorded on exit from this menu. Otherwise, an exit is made to the main MENU with the instrument baseline settings intact, a feature of the CARY which prevents inadvertent alterations to the parameters describing the current instrument Baseline.

'I'.....Instrument Settings

This selection branches to label INSTRUMENT, reads the current instrument settings and presents the most important in a menu arrangement similar to the equivalent display on the CARY. The user may alter these instrument settings and any changes are implemented immediately by the instrument. If such changes affect the quality of the Baseline matching for a subsequent acquisition scan then the changes will be overridden automatically, if possible. Otherwise, the user will be directed to record a new Baseline scan with the altered settings, followed by acquisition of the spectrum. In most cases the automatic matching routines will take effect to provide a smooth user interface.

'L'.....Lamps/Detectors/Accessories

This selection branches to label ADVANCED and reads the current instrument settings and presents a number of menu selections for subsidiary functions and operating modes of the CARY. Selection '1' branches to label LAMP and presents a menu which lists the status of the lamp and detector modes, which then may be altered. Selection '2' branches to label ACCESSORY and presents a menu which lists the status of the temperature and printer accessories, which then may be setup as desired. Selection '3' for automatic operations is not yet supported.

'S'.....Store File On Disk

This selection branches to label STORE and prompts the user for entry of pertinent file information before saving a data file to disk. This routine includes standard error checking for File Exists, File Open and disk transfer errors. The user is returned to the main MENU on exit.

'X'.....Exit

This selection braches to label EXIT and checks for the presence of an unstored spectrum which causes a prompt for confirmation before proceeding. The user then has the option of setting the CARY to standby mode, if desired, before the program stops.

SPECTRUM:

This portion of CARYSPEC controls the acquisition of a spectrum from the CARY 2390 spectrophotometer. On entry, this code will check important instrument parameters and status variables and perform conditional branches to BASELINE, INSTRUMENT or MENU if the conditions outlined above are not satisfied. A successful entry will display a request for the wavelength scan limits, which default to the Baseline scan range. New limits may be chosen and are validated for the range 185–3152 nm. {The limits may exceed the Baseline range but this will cause a subsequent call to SUBROUTINE Bline with the new limits and current instrument settings before returning to the data acquisition loop.}

The remaining entry required is the step size interval (0.01–5 nm) during the scan. The instrument is capable of 0.01 nm steps in the UV–VIS region or 0.04 nm in the Near IR. No restrictions are placed on the user in this regard but it is *strongly recommended* that sensible units be chosen, *e.g.* .1, .2, .25, .5 nm. The program will reject combinations of scan rate and step size which would result in the data rate exceeding 5 Hz. This restriction is a result of the rather slow multi-user environment of the HP 1000 rather than a processing speed problem. Either scan rate or step size may be altered to meet this condition. Finally, the wavelength range and step size are used to check the number of data points for the scan. If the request exceeds 10001 points the user is prompted for a new step size.

After satisfying the basic conditions above the program will perform a number of checks on the current operating conditions of the CARY 2390 to determine whether these will match the conditions for the Baseline scan. Mismatched settings of SBW (nm) and GAIN will be reset automatically to smooth over some instrument peculiarities. Other mismatches are assumed to be operator requirements and result in a prompt to record a new Baseline scan. The user may either proceed or abort this operation and return to the main MENU to take corrective action.

Successful traversal of all the matching checks will present a listing of scan parameters and a prompt to Start or Abort the scan. Aborting will return the user to the main MENU and restore the parameter strings describing any previous spectrum in memory. Starting will position the monochromator to the starting wavelength and prompt for Print to Screen during the scan – removal of this I/O overhead helps prevent missed data with several users on the HP 1000. The remainder of the acquisition is automatic, returning to the main MENU after completion.

BASELINE:

The current instrument Baseline parameters are read on entry to this section of CARYSPEC for display in a menu format similar to the Baseline Setup menu on the instrument. This code is responsible for the selection of all relevant parameters for a new Baseline scan. However, since most of these parameters are masked from direct changes by the computer, via SUBROUTINE Select, a number of inappropriate combinations are tested for after each new selection is made. These tests reset the bad parameter requests to the most appropriate selection thereby eliminating the rejection of any parameters in subsequent calls to SUBROUTINE Bline. After validation of the Baseline parameters the program tests whether the operating mode at the start of the scan will be AUTO GAIN (MODE = 1) or AUTO SLIT (MODE = 2). The integer variable MODE then controls the logical operation of the remainder of the program and SUBROUTINE Bline where choices between SBW and GAIN settings are important for determining or controlling the operation of the CARY 2390.

A special exit is made from the BASELINE code for setting the instrument GAIN level if the requested value exceeds the current setting by more than a factor of 10, which can result in misbeaviour of the slit servo system of the CARY 2390. A sudden, large increase in GAIN should just send the instrument closer to zero slit width. However, on *this* CARY 2390, at least, the slit width can overshoot through zero and continue to fully open the slits with high gain, seriously imperilling the detectors! To prevent such a disaster the program will select AUTO SLIT mode and branch to line 490, which is part of the INSTRUMENT code fragment. This subsection resets the current instrument GAIN in factors of 10 until it matches the new Baseline request. The logical variable TRANSFER controls the exit from this routine back to BASELINE.

On exit from the Baseline Setup menu the user may either record a new Baseline or return to the main MENU. Both options read the current instrument parameters before returning to MENU, keeping the program updated. This is performed by re-using part of the code at the start of the BASELINE fragment under the control of logical variable TRANSFER. If a new Baseline is recorded the program will monitor the instrument until completion of the procedure and then issue an AUTO BALANCE command to zero the instrument on the reference material. Subsequent data acquired via the SPECTRUM fragment will therefore produce baseline corrected spectra.

ADVANCED:

This portion of CARYSPEC presents a small menu of subsidiary instrument setup functions that may need to be changed occasionally. The selection are:

'1'.....LAMPS & DETECTORS

This selection will display a further menu which lists the current status of the LAMP POWER, LAMP SELECT and DETECTOR SELECT modes. Normally, both lamps are ON and the lamps and detectors are in AUTO SELECT mode. These settings can be changed to increase the working wavelength range for the individual lamps or detectors. {NOTE: Individual selection of a lamp or detector prevents lamp or detector changes and thereby prevents coverage of part of the wavelength range accessible with AUTO SELECT modes.}

'2'.....ACCESSORY SETTINGS

This selection allows the user to turn on and setup two installed accessories, the TEMPERATURE READOUT and the thermal PRINTER. On entry to this routine both accessories are commanded to an OFF status. If a positive reply is received from the CARY that parameter is reset to ON. If the TEMPERATURE accessory is selected and turned ON a small menu is presented for selection of the TEMPERATURE RANGE. If the PRINTER option is selected and turned ON a subsidiary menu is presented to select the operating mode and interval step size for printer output. The modes supported are Wavelength, Time and Temperature. However, CARYSPEC only acquires data in Wavelength mode.

'3'.....AUTO OPERATIONS

This selection is intended for future expansion for automatic repetitive scans. Currently, it prints an error message and returns for another selection.

'X'.....EXIT TO MENU

This entry returns to the main MENU.

INSTRUMENT:

This section of CARYSPEC reads the current wavelength and instrument settings from the CARY and presents the most important functions in a menu forn.at that is very similar to the equivalent display on the instrument. Changes made from this menu are executed by the CARY 2390 immediately. The selections are:

'0' .....WAVELENGTH

This selection allows the monochromator to be repositioned to any valid wavelength for the current selections of Lamp and Detector modes.

'1' .....ORDINATE

Only Absorbance, %T and Temperature are selectable from this menu. However, CARYSPEC will not allow Temperature as a valid ordinate during scans.

'2' .....ABSCISSA

Wavelength, Time and Temperature are selectable from this menu. However, CARYSPEC only allows Wavelength as a valid abscissa during scans.

'3' .....SCAN RATE

The scan rate must be chosen in combination with spectral bandwidth and filter period for accurate recording of bandshape. There is a particular difficulty in the 650 nm region where a Wood's anomaly causes poor baseline correction. The scan rate should not exceed 1 nm/sec per SBW (nm) per second period.

'4' .....CHART DISPLAY

The chart recorder may be used on any setting during data acquisition.

'5' .....REFERENCE MODE

The instrument is normally used in AUTO SELECT mode to allow full wavelength coverage with both lamps and both detectors. However, AUTO GAIN and AUTO SLIT modes may also be used for wavelength scans. The working range for these depends on the detector mode selected. AUTO GAIN may be used above 800 nm with the PM Tube if the UV/VIS detector modes is selected. AUTO SLIT mode can be used for the full instrument range (185-3152 nm). The reference mode for data acquisition must match that used for the Baseline scan. SINGLE BEAM mode is not valid for wavelength scans and is intended only for instrument adjustments.

'6' .....SBW (nm), GAIN

This selection allows setting of *either* the SBW or GAIN depending on whether the CARY is operating in AUTO GAIN or AUTO SLIT mode at the current wavelength. The actual operating mode for AUTO SELECT reference mode is determined by the wavelength and detector select mode. These are checked by CARYSPEC to determine the correct prompt and instrument command.

'7' .....PEN FUNCTION

The pen operates independently of the raw spectrophotometric data sent via the IEEE-488 bus to an external computer and may operate in any valid mode during data acquisition. The Ordinate choice determines which modes are valid and inappropriate selections are masked by CARYSPEC. However, it is perfectly feasible to draw a second derivative spectrum while acquiring data via the IEEE-488 bus.

'8' .....PEN LIMITS

This selection allows for setting the range and offset for all valid Ordinate modes. The NORMAL mode pen limits are selected via the Parameter and Variables Tables. The %T mode has a minor bug for the 200 %T scale. If a previous choice has set a non zero offset this will not be correctly reset to 0 %T as expected since the $\%T_{zero}$ variable is masked off by the CARY on the 200 %T scale. Similarly, the Temperature zero offset parameter *can not be changed* by an external computer, though the setting *is read* by CARYSPEC. A non-zero offset entered from the instrument keypad will be displayed but only the range can be set by CARYSPEC. The Derivative and Log(Abs) mode limits are handled by parameter table selection with special handling of the indexing to prevent use of invalid settings in the CARY firmware table.

'9' .....RESPONSE TIME (sec)

This selection allows the filter period to be set to 0.5, 1, 3 or 10 seconds.

During the recording of a Baseline the period should be set to 0.5 seconds for maximum fidelity in the 650 nm region where there is a Wood's anomaly. Failure to observe the scan rate, filter period and SBW limitations will results in improper baseline corrections. Higher period settings can be used on subsequent spectra with little prejudicial effect.

'10'.....BEAM INTERCHANGE

This selection allows the front and rear light beam paths to be interchanged between the sample and reference channels for special applications, such as the diffuse reflectance accessory.

'11'.....SLIT HEIGHT

This selection is not valid - it is a READ ONLY parameter for the manual slit height setting.

'X' .....EXIT Instrument Menu

This selection performs a return to the main MENU.

LAMP:

This subsidiary menu reads and reports the current status of the lamp and detector operating modes. The selections are:

'1' .....LAMP POWER

Normally, the instrument is operated with this parameter set to BOTH ON enabling the complete wavelength range to be covered. However, the UV or VIR/NIR selections may be made to prolong the life of a lamp. CARYSPEC *does not* automatically turn on lamps as required for a particular scan.

'2' .....LAMP SELECT

For complete coverage of the wavelength range 185–3152 nm this parameter must be in the AUTO select mode, which will result in a lamp change at 340 nm. The range covered by the individual lamps may be extended - up to 400 nm for the D_2 lamp (UV) and down to 270 nm for the tungsten lamp (VIS/NIR). However, no lamp change will then be made. CARYSPEC provides error checking for the latter two modes to prevent positioning the monochromator outside the valid wavelength limits since this would result in the instrument turning OFF the current baseline. Recovery from such a state involves repositioning the monochromator and using the ON/SETUP selection for Baseline Status in the Baseline Setup menu.

'3' .....EXIT TO MENU

This selection returns to the ADVANCED menu.

ACCESSORY:

The CARY 2390 has two installed accessories programmable by an external computer. The selections are:

'1' .....TEMPERATURE READOUT

This selection should normally be turned ON so that subsequent data files are stored with the correct temperature (a reading of 2.55 is passed by the CARY with the accessory OFF). If this selection is made and turned ON a further menu will be presented for selection of the temperature range. This only affects the pen scaling with TEMPERATURE as the Ordinate or Abscissa – not valid modes for data acquisition in CARYSPEC. Thus, selecting 100 degrees is recommended.

'2' .....PRINTER

This selection allows the user to setup the thermal strip printer to provide instrument readings at selected intervals during a scan (1 point/sec max.). While the selections include WAVELENGTH, TIME and TEMPERATURE the latter two are not valid scan modes in CARYSPEC.

STORE:

This portion of CARYSPEC provides the data file storage routine. On entry to this code CARYSPEC checks that a spectrum has been acquired and has not yet been stored. Otherwise, the program returns to the main MENU. After this validation the program will present a series of prompts for Filename and Directory information, which are then used to build a Pathname and to check that such a file does not already exist in the specified directory. If the filename is valid the user will be prompted for the LABEL, DATE, CONCENTRATION (M) and PATHLENGTH (cm) file descriptors followed by disk file storage. The data file is stored in the following format given in Table II.

EXIT:

This final portion of CARYSPEC checks that any spectrum in memory has been stored and prompts for confirmation before allowing the user to terminate the program. On exit the user may elect to set the CARY 2390 to standby mode if no further spectra will be acquired.

TABLE II

Data File Format

Line	File Variables	Format Type [a]
1	TITLE	CHARACTER (A72)
2	DATE	CHARACTER (A8)
3	XMIN,XMAX,XSTEP,CONC,PATH	REAL (*)
4	ORD,ABSC,CELL,CYCLE,SAMPLE, WAVE,TIMER,TEMP,DIST	REAL (*)
5	J,K,NARRAY	INTEGER (I3,I3,I6)
6–54	PARAM(I)	INTEGER (I2)
55	VARIABLE(I)	REAL (*)
56–/	Y(I)	REAL (*)
/–eof	X(I)	REAL (*)

a: (*) indicates free field format

2.3 BLOCK DATA:

All COMMON variables used by CARYSPEC are held in named COMMON blocks and initialized in BLOCK DATA immediately following the main program unit. The compiler directive /NOALLOCATE/ is used to ensure that only one block of memory is set aside during the multi-level segmentation of CARYSPEC. The variables contained within the COMMON blocks are listed below:

/MODE/ Contains COMMON INTEGER variables

NDATA Number of data points in a scan
Value set in main program
Value used in SUBROUTINE Acquire
Value stored in NARRAY in main program for disk data file

XMODE Specifies abscissa mode for selecting length of data field
Value set in main program
Value used in SUBROUTINE Acquire as index for local array XOFF

YMODE Specifies ordinate mode for selecting length of data field
Value set in main program
Value used in SUBROUTINE Acquire as index for local array YOFF

/CARY/ Contains COMMON REAL variables

ORD, ABSC, CELL, CYCLE, SAMPLE, WAVE, TIMER, TEMP, DIST
Values correspond to the 9 instrument readings sent during scan
Values set in SUBROUTINE Acquire
Values used in main program
Values stored in disk data file

/IP/ Contains COMMON INTEGER arrays

NPAR Specifies the number of settings for each instrument parameter
DIMENSION = 49
Values set in BLOCK DATA
Values used by SUBROUTINE Select for changing instrument settings

OFFSET Species the index offset in the parameter table
DIMENSION = 49
Values set in BLOCK DATA
Values used in SUBROUTINE Select for changing instrument settings

/IS/ Contains COMMON CHARACTER string arrays

Pname Specifies the names of instrument parameters for screen display
DIMENSION = (49)*10 characters
Values set in BLOCK DATA
Values used in SUBROUTINE Select

Vname Specifies the names of instrument variables for screen display
DIMENSION = (49)*8 characters
Values set in BLOCK DATA
Values not used in current version (for future use)

2.4 SUBROUTINES:

CARYSPEC uses subroutines to perform specific tasks which, with one exception, are required more than once. Terminate is setup as a subroutine solely for clarity of main program execution. The purpose and calling sequences are listed below:

Acquire(Inc,PRINT,SINGLE,WAVELENGTH)

Performs real time data acquisition from the CARY 2390 in two modes, single point for updating the current monochromator position and scan mode at Inc (nm) steps. COMMON variables XMODE, YMODE and NDATA select the correct data string format for the Abscissa and Ordinate modes selected by the main program unit and the number of data points required in the scan. The scan mode stores each Abscissa and Ordinate value in EMA COMMON arrays X and Y. COMMON BLOCK /CARY/ returns the final set of readings to the main program unit for data file storage.

/MODE/ INTEGER XMODE,YMODE,NDATA input variables control acquisition

/CARY/ REAL ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST output

/DATA/ REAL arrays X,Y hold Abscissa and Ordinate values for output

Inc CHARACTER*4 variable input which specifies the interval (nm)

PRINT LOGICAL variable input which turns screen output on/off during scan

SINGLE LOGICAL variable input which selects single datum or scan mode

WAVELENGTH
REAL variable output for single datum mode

CALLED BY: *Main program unit only*
CALLS: SUBROUTINE Val

Bline(WMIN,WMAX,Bdet,Bgain,Blamp,Bperiod,Brate,Bref,Bsbw,Bslit.Bscan,Btime.
MATCH,MODE)

Performs a Baseline Setup by sending a list of instrument parameter requests to the CARY 2390. Validation of the instrument settings is performed by the main program unit before calling Bline.

WMIN,WMAX	Wavelength limits passed from main program unit
Bdet	CHARACTER*1 variable input to select detector mode
Bgain	CHARACTER*4 variable input to set gain value
Blamp	CHARACTER*1 variable input to select lamp mode
Bperiod	CHARACTER*1 variable input to select period setting
Brate	CHARACTER*1 variable input to select scan rate setting
Bref	CHARACTER*1 variable input to select reference mode
Bsbw	CHARACTER*4 variable input to set SBW value
Bslit	CHARACTER*1 variable input to match physical slit height
Bscan	CHARACTER*14 variable input for screen display of scan rate
Btime	CHARACTER*14 variable input for screen display of period
MATCH	LOGICAL variable .TRUE. on entry and exit unless scan aborted
MODE	INTEGER variable input to specify AUTO GAIN/SLIT mode

Center(TITLE)

Prints a string on the user console centred within a 72 column line.

TITLE CHARACTER*72 string, contents set by calling unit

CALLED BY: Main program unit, SUBROUTINE Select, SUBROUTINE Bline and SUBROUTINE Acquire

CALLS: None

GOTO(Wlength)

Performs the same function as the Key Pad GO TO WLNGTH on the instrument to enable repositioning of the monochromator to a specified wavelength. Error trapping for illegal or inappropriate settings is performed by the main program unit and no negative reply is tested for.

Wlength CHARACTER*7 variable input from the main program unit

CALLED BY: Main program unit only

CALLS: SUBROUTINE Send, SUBROUTINE Instats

EXTENSION: LEN(*string*) function, HP extension to FORTRAN 77

Instats(Slew,...,Windex)

Performs a request for current instrument status from the CARY 2390. Slew is used to determine if the monochromator is still in motion. The other variables are not used in this version. No negative reply is issued by the CARY for this command.

Slew,Model,Ncell,Range,Windex

CHARACTER*1 variables passed back to calling unit

CALLED BY: Main program unit, SUBROUTINE GOTO

CALLS: None

Limits(MIN,MAX)

Reads entries for the wavelength limits from the user console, swaps the entries if necessary and validates the entries against the instrument limits (185 - 3152 nm). The values are then rounded to whole digits.

MIN,MAX REAL variables passed back to main program unit

CALLED BY: Main program unit only
CALLS: None

Line(N)

Prints a line of '–' characters to the user console N columns wide and centred within a 72 column line.

N INTEGER variable input from calling unit

CALLED BY: Main program unit, SUBROUTINE Select, SUBROUTINE Bline and SUBROUTINE Acquire
CALLS: None

Partable(PARAM)

Performs a request to send the parameter table from the CARY 2390 and processes the reply to update the program's iist of current instrument settings held in the integer array PARAM. No negative reply is issued by the CARY for this command.

PARAM INTEGER array output which holds the instrument parameter settings DIMENSION = 49, values set by CARY and SUBROUTINE Select

CALLED BY: Main program unit only
CALLS: None

Select(N,PARAM,Pstr)

Performs selection of available instrument settings for parameter N. Calls SUBROUTINE Send(Command) to set new parameter values. Negative replies are not tested since the parameter table values are read again on return to the main program menus calling Select. A special fix has been added for Derivative modes to use only valid selections from PARAM(11) and Pstr(11).

N INTEGER input value (1 - 49) representing parameters 0 - 48

PARAM INTEGER array input of current instrument parameter settings
DIMENSION = 49, used to detect special case indexing for Pstr

Pstr CHARACTER string array containing all selections for parameters
DIMENSION = (49,16)*14 characters, 41-49 not used in this version

/IP/ INTEGER arrays NPAR,OFFSET used to select index number for Pstr

/IS/ CHARACTER array Pname containing the names of each parameter
DIMENSION = (49)*10

INDEX INTEGER array of valid index values for Derivative modes
DIMENSION = 11, uses local data for indices to PARAM(11)

Send(Command)

Performs an IEEE-488 WRITE to the CARY 2390 to send a string command to the instrument and to read the reply. Negative replies are not checked using this routine. Commands are either validated before calling Send or parameters and variables are read afterwards to check the results from Send.

Command CHARACTER variable holding an ASCII string command for CARY
DIMENSION = variable, set by calling unit.

CALLED BY: Main program unit, SUBROUTINE Select, SUBROUTINE GOTO
CALLS: None

Str(VALUE,String,PREC)

Performs a conversion from numeric value to a string number for floating point numbers only with up to 12 digits precision. This is more than required by the CARY 2390.

VALUE REAL variable input to be processed by the routine

String CHARACTER*14 string output corresponding to VALUE

PREC INTEGER variable input to set the rounding precision for string

CALLED BY: Main program unit, SUBROUTINE Bline
CALLS: None

Terminate

Performs an IEEE-488 WRITE to UNTALK the CARY 2390 and terminate real time transmission mode.

CALLED BY: Main program unit only
CALLS: ABRT(35,3) an EXTERNAL class system level routine
This function sends the UNTALK/UNLISTEN characters '_?'

Upper(Code)

Performs a check for lower case characters in a string of arbitrary length and converts to upper case if necessary.

Code CHARACTER variable passed into routine and UPPER case on exit
DIMENSION = arbitrary, set by calling unit

CALLED BY: Main program unit, SUBROUTINE Bline
CALLS: None

Val(String,VALUE)

Performs a conversion from string to numeric value for a string number containing up to 10 digits. This is more than required by the CARY 2390.

String CHARACTER string input to be processed by routine
DIMENSION = arbitrary, set by calling unit

VALUE REAL variable output

CALLED BY: Main program unit, SUBROUTINE Vartable, SUBROUTINE Acquire
CALLS: None

Vartable(VARIABLE)

Performs a request to send all 14 instrument variables and processes the replies to update the program's list of current values held in the floating point array VARIABLE. Negative replies from the CARY are not tested in this routine since illegal requests are not issued by Vartable.

VARIABLE REAL array output which holds the instrument operating variables
DIMENSION = 14, values set by CARY and main program unit

CALLED BY: Main program unit only
CALLS: SUBROUTINE Val

Wait(DELAY)

Performs a loop which tests the system clock until DELAY seconds have elapsed. The routine does not make provision for the special case at the transition to 2400 hours.

DELAY REAL variable holding the value of the delay period in seconds

CALLED BY: Main program unit and SUBROUTINE Bline
CALLS: FUNCTION Time(I)

2.5 FUNCTIONS:

CARYSPEC uses only one function subprogram that makes an EXEC call to read the system time.

Time(I)

Performs an EXEC call to read the system clock and converts the reading to seconds and centiseconds.

I	Dummy argument
CALLED BY:	SUBROUTINE Wait only
CALLS:	EXEC(ICODE,ITIME) system level command

PROGRAM CODE

3.0 Source Code Availability:

The source code for program CARYSPEC is an 83K ASCII text file available on either a Hewlett-Packard cartridge tape or an IBM 360K format floppy disk. All requests should be accompanied by the blank medium desired. A printed copy of the source code is listed below.

3.1 Variable Names And Usage:

A complete listing of the INTEGER, REAL and CHARACTER variables for the MAIN segment of CARYSPEC is given below in Tables III, IV & V, respectively. The subroutines use the same names as the main program for the same variables. Additional variables in the subroutines and simple integers, I-N, are not documented since their usage is rather obvious. The logical variables MATCH, PRINT, SINGLE and TRANSFER are used within the program to control conditional branching. MATCH is related to BLOCK IF tests for matching of the baseline and spectrum parameters. PRINT controls whether data will be printed to the console screen during data acquisition. SINGLE controls the operation of Acquire to update the wavelength. TRANSFER is used for special branching to reuse portions of code.

Table III

Glossary of INTEGER Variables

Name	Description	Value
ACCESSORY	Assigned Label - Accessory Setup Menu	600
ADVANCED	Assigned Label - Subsidiary Functions	350
BASELINE	Assigned Label - Baseline Setup Menu	200
EXIT	Assigned Label - Terminate Program	900
INSTRUMENT	Assigned Label - Read Cary Settings	390
LAMP	Assigned Label - Lamp & Detector Modes	570
MENU	Assigned Label - Main Control Menu	10
PARAMETERS	Assigned Label - Instrument Setup Menu	400
SPECTRUM	Assigned Label - Acquire Spectrum	90
STORE	Assigned Label - Store Disk File	700
ASCII	ASCII code for a command output	> 48
MODE	Controls the selection of AUTO GAIN/SLIT	0,1
NARRAY	Number of data points in spectrum - file	1-10001
NCOL	Number of screen columns in menu display	50-70
NDATA	Number of data points in spectrum - Acquire	1-10001
PREC	Precision for rounding function in Str	3,4
XMODE	Selects abscissa data format in Acquire	1
YMODE	Selects ordinate data format in Acquire	1,2
NPAR(49)	Number of settings for each parameter	1-16
OFFSET(49)	Index offset for parameter settings	0-11
PARAM(49)	Instrument operating modes table	1-16

Table IV

Glossary of REAL Variables

Name	Description
ORD	Final ordinate value returned by Acquire
ABSC	Final abscissa value returned by Acquire
CELL	Final cell # value returned by Acquire
CYCLE	Final cycle # value returned by Acquire
SAMPLE	Final sample # value returned by Acquire
WAVE	Final wavelength value returned by Acquire
TIMER	Final time value returned by Acquire
DIST	Final distance value returned by Acquire
BAND	Spectral Bandwidth (nm) - AUTO GAIN mode
CONC	Concentration of sample (M) - file variable
GAIN	Instrument gain - AUTO SLIT mode
NUMBER	General purpose data entry variable
PATH	Pathlength of sample cell (cm) - file variable
PMIN	Pen scale minimum limit
PMAX	Pen scale maximum limit
RATE	Numeric equivalent of scan rate parameter
RATIO	Variable for data rate & slit gain checks
SPECBAND	File variable for SBW (nm) at λ_{min} (nm)
SPECGAIN	File variable for GAIN at λ_{max} (nm)
STEP	Numeric value of step size (nm) interval
WAVELENGTH	Current monochromator position (nm)
WMIN	Requested ending wavelength for scan
WMAX	Requested starting wavelength for scan
XMIN	File variable for WMIN
XMAX	File variable for WMAX
XSTEP	File variable for STEP
ZERO	Pen scale offset variable, %T and Deriv. modes
VARIABLE(14)	Instrument operating conditions table
X(10001)	Wavelength array
Y(10001)	Absorbance or %T array

Table V

Glossary Of CHARACTER Variables

Name	Description
Screen Control:	
BELL	CHAR(7) bell character
CLR*2	Clear screen
DOWN*2	Move cursor down 1 line
ESC	CHAR(27) escape character
HOME*2	Move cursor to upper right corner
UP*2	Move cursor up 1 line
Instrument Status:	
Bdet	Baseline detector mode
Bgain*4	Baseline gain setting
Bgbw*4	Baseline SBW or GAIN, depending on mode
Blamp	Baseline lamp mode
Bmin*4	Baseline ending wavelength
Bmax*4	Baseline starting wavelength
Bperiod	Baseline filter setting
Brate	Baseline scan rate
Bsbw*4	Baseline spectral bandwidth
Bslit	Baseline slit height
Odet	Previous spectrum detector mode
Ogain*4	Previous spectrum gain setting
Olamp	Previous spectrum lamp mode
Omin*4	Previous spectrum ending wavelength
Omax*4	Previous spectrum starting wavelength
Operiod	Previous spectrum filter setting
Orate	Previous spectrum scan rate
Osbw*4	Previous spectrum spectral bandwidth
Oslit	Previous spectrum slit height
Sdet	Spectrum detector mode
Sgain*4	Spectrum gain setting
Slamp	Spectrum lamp mode
Smin*4	Spectrum ending wavelength
Smax*4	Spectrum starting wavelength
Speriod	Spectrum filter setting
Srate	Spectrum scan rate
Ssbw*4	Spectrum spectral bandwidth
Sslit	Spectrum slit height

Instrument Control:

Accon*3	Turn accessory on command
Accoff*3	Turn accessory off command
Autobal*4	Perform auto balance to zero reading
Command*44	String of instrument commands to CARY
CSM	Checksum character for data transmission
Blstat*5	Read baseline status parameter command
Instr*4	Recall instrument setup menu display
Key*2	Press key command
Lock*4	Lock keyboard command
Messon*3	Send message to line 4 of CARY display
Messoff*4	Clear message from CARY display
Par*3	Read parameter command
Parset*2	Change parameter command
Ready*4	Release CARY from standby mode
Response*64	String for reply messages from CARY
Setup*4	Record trailer setup command
Standby*4	Place CARY in standby mode
Start*4	Issue a start command
Stop*4	Issue a stop command
String*14	String to pass data to or from subroutines
Unlock*4	Unlock keyboard command
Var*3	Read variable command
Varset*3	Change variable command

Program Control:

Access(5)*4	Accessory status (OFF, ON)
Astat*10	Auto balance status (OFF, SET)
Bscan*14	Scan rate, Baseline screen output
Bstat*10	Baseline status (OFF, ON, ON/MATCH)
Btime*14	Filter period, Baseline screen output
Code	Menu selection variable
Icode	Parameter setting in ASCII format
Pcode	Parameter number in ASCII format
Pname(49)*10	Table of names for each parameter
Printer(6)*12	Printer operating mode
Pstr(49)*14	Table of names for each parameter setting
Sstat*10	Spectrum status (OFF, ACQUIRED, STORED)
TITLE*72	String to be printed to screen
Vname(14)*8	Table of names for each variable
Wlength*7	Wavelength in ASCII format for GOTO

File Storage:

DATE*8	Date in mm/dd/yy format
Directory*40	Directory pathname
Fname*20	Filename and extension
INITIALS*2	User's initials for extension .Sxx
Name*16	Filename without extension
Outfile*63	Complete pathname for file

```
FTN7X,L
$FILES 0,1                      ! # Setup One Disk I/O File
$ALIAS /MODE/,NOALLOCATE        ! # BLOCK DATA Holds Values Of Named
$ALIAS /CARY/,NOALLOCATE        !   COMMON Variables So Don't Allocate
$ALIAS /IP/,NOALLOCATE          !   Memory For These Here - SGMTR Will
$ALIAS /IS/,NOALLOCATE          !   Create Memory For These As Required
$EMA /DATA/                     ! # Use EMA Space For Large Data Arrays
C
C     ****************************************************************
C
      PROGRAM CARYSPEC
C
C     ****************************************************************
C
C     This Program Is Designed To Control Data Acquisition From The
C     CARY 2390 UV-VIS-NIR Spectrophotometer Via The IEEE-488 Bus:
C
C     The CARY 2390 Is Addressed As Device #3 On The IEEE-488 Bus.
C
C     The HP 1000 Is Configured To Operate The IEEE-488 Bus In ASCII
C     Data Record Mode With Auto Addressing Enabled. The Bus Occupies
C     Logical Unit Addresses 35 - 38 (Device Addresses 0 - 3). LU 38
C     Controls The CARY 2390 And LU 35 Is Used To Issue Bus Commands.
C
C     ----------------------------------------------------------------
C
C
C         AUTHOR:  Dr. Robert A. Binstead,
C                  Chemistry Division, Code 6125,
C                  Naval Research Laboratory,
C                  Washington. D.C. 20375-5000
C
C
C         WRITTEN: December, 1986 - January, 1987
C
C         VERSION: 1.7
C
C         REVISED: March, 1987:
C                                 - Modified to store Abscissa (X) array
C                                   after Ordinate (Y) values to prevent
C                                   data file corruption in the event of
C                                   missed data points during multiuser
C                                   sessions where the HP 1000 can not
C                                   keep up with data transmission rate.
C                  May, 1987:
C                                 - Modified MATCHING criteria between
C                                   Spectrum & Baseline to omit checks
C                                   on Scan Rate & Period. This allows
C                                   the Baseline scan to be taken under
C                                   conditions for the best correction
C                                   of instrumental artifacts.
C                                 - On MISMATCHED BASELINE detection the
C                                   program will collect a new baseline
C                                   with instrument parameters specified
```

```
C                                for the spectrum except for PERIOD
C                                and SCAN RATE which revert to those
C                                for the previous baseline scan.
C                              - Altered data storage routine to use
C                                default or specified cartridge #.
C                   June, 1987:
C                              - Modified Filename convention to match
C                                the use of Directory Paths in the
C                                new CI operating system.
C                              - Segmented the program using SGMTR
C                                to fit within CI's smaller boundary.
C                   August, 1987:
C                              - Eliminated INQUIRE statement for FILE
C                                EXISTS or FILE OPEN check since this
C                                caused a memory protect error in the
C                                segmented versions if the filename
C                                was already in use. These checks are
C                                made using the IOSTAT number returned
C                                by the OPEN statement instead.
C                   November,1987:
C                              - Altered updating of Variables Table
C                                so that SBW at Smin and GAIN at Smax
C                                are stored in Data File.
C                   January, 1988:
C                              - Placed All COMMON Variables In Named
C                                COMMON Blocks To Prevent Them From
C                                Being Re-initialized On Calls To Other
C                                Nodes Of The Multi-Level Segmentation.
C                              - Explicitly Specified Allocation Of
C                                BLOCK DATA memory Using NOALLOCATE
C                                Compiler Directives.
C                              - Eliminated Overwriting Of Data File
C                                Variables By The Wavelength Reading
C                                Routine. The CALL To Acquire Has
C                                Been Augmented To Bypass The Usual
C                                Spectral COMMON Variables in /CARY/.
C                   February,1988:
C                              - Removed Single Beam Operation Since
C                                The Cary Cannot Acquire A Baseline
C                                In This Operating Mode.
C                              - Added Tight Checking For Improper
C                                Combinations Of Baseline Detector,
C                                Lamp and Reference Mode Requests.
C                              - Revised AUTO GAIN vs AUTO SLIT Mode
C                                Selection In Baseline And Instrument
C                                Setup Menus To Utilize Detector Mode
C                                Under AUTO SELECT Reference Selection.
C                              - Added Automatic Adjustment Of SBW And
C                                GAIN Before Scan To Match Baseline.
```

```
C
C        MODES:    All Abscissa & Ordinate Modes (SUBROUTINES)
C                  Absorbance or %T vs Wavelength (PROGRAM only)
C
C        MEMORY:   28,000 Words (Max.PATH) + 40,000 Words EMA (DATA)
C                   3,000 Words (MSEG)     + 5 Memory Resident Nodes
C                        - 83 Page Partition Required -
C
C        SEGMENT:  This Program Is Too Large To Run Under CI On The
C                  HP 1000 - It Must Be Segmented Using SGMTR And
C                  MLLDR Loader - A CMD Transfer File SEGMCARY.CMD
C                  Contains The Commands To Accomplish This.
C
C     ----------------------------------------------------------------
C
      INTEGER ACCESSORY,ADVANCED,BASELINE,EXIT,INSTRUMENT
      INTEGER LAMP,MENU,PARAMETERS,SPECTRUM,STORE
      INTEGER ASCII,MODE,NARRAY,NCOL,NDATA,PREC,XMODE,YMODE
      INTEGER NPAR(49),OFFSET(49),PARAM(49)
      REAL ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST,BAND,CONC
      REAL GAIN,NUMBER,PATH,PMIN,PMAX,RATE,RATIO,SPECBAND,SPECGAIN
      REAL STEP,WAVELENGTH,WMIN,WMAX,XMIN,XMAX,XSTEP,ZERO
      REAL VARIABLE(14),X(10001),Y(10001)
      LOGICAL MATCH,PRINT,SINGLE,TRANSFER
C
C        Dimension Screen Control String Variables
C
      CHARACTER BELL,CLR*2,DOWN*2,ERASE*2,ESC,HOME*2,UP*2
C
C        Dimension Instrument Control String Variables
C
      CHARACTER CSM,Lock*4,Unlock*4,Key*2,Accon*3,Accoff*3
      CHARACTER Par*3,Var*3,Parset*2,Varset*2,Messon*3,Messoff*4
      CHARACTER Setup*4,Command*44,Response*64,String*14
C
C        Dimension Specific Key or Function String Variables
C
      CHARACTER Ready*4,Standby*4,Start*4,Stop*4,Instr*4,Autobal*4
C
C        Dimension Program Parameter Variables
C
      CHARACTER Sstat*10,Bstat*10,Astat*10,Wlength*7
      CHARACTER Directory*40,Fname*20,Name*16,Outfile*63
      CHARACTER Smin*4,Smax*4,Sinc*4,Sdet,Sgain*4,Slamp,Speriod
      CHARACTER Srate,Sref,Sslit,Ssbw*4
      CHARACTER Bmin*4,Bmax*4,Bdet,Bgain*4,Blamp,Bperiod
      CHARACTER Brate,Bref,Bslit,Bsbw*4,Fperiod,Frate
      CHARACTER Omin*4,Omax*4,Oinc*4,Odet,Ogain*4,Olamp,Operiod
      CHARACTER Orate,Oref,Oslit,Osbw*4
C
      CHARACTER DATE*8,INITIALS*2,TITLE*72
      CHARACTER Access(5)*4,Printer(6)*12,Code,Icode,Pcode
      CHARACTER Pname(49)*10,Vname(14)*8,Bscan*14,Btime*14
      CHARACTER Pstr(49,16)*14
```

```
C
C     ------------------------------------------------------------------
C
      COMMON /MODE/NDATA,XMODE,YMODE
      COMMON /CARY/ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST
      COMMON /IP/NPAR,OFFSET,/IS/Pname,Vname
      COMMON /DATA/Y,X
C
C         Reference Library IEEE-488 Subroutines
C
      EXTERNAL ABRT
C
C     ------------------------------------------------------------------
C
C         Initialize Cary Command & Status String Variables
C
      DATA CSM,Setup,Lock,Unlock,Key/'0','@@20','@A10','@A00','@D'/
      DATA Accon,Accoff,Par,Var/'@F1','@F0','@C1','@C2'/
      DATA Parset,Varset,Messon,Messoff/'@H','@I','L1','@L00'/
      DATA Ready,Standby,Start/'@DT0','@DS0','@DP0'/
      DATA Stop,Instr,Autobal/'@DQ0','@DC0','@DU0'/
      DATA Sstat,Bstat,Astat/'  OFF','  OFF','  OFF'/
C
C         Initialize Cary Instrument Settings String Arrays
C
      DATA (Pstr(1,I),I=1,6)/'ABSORBANCE','% TRANSMISSION',
     &'TEMPERATURE','% REFLECTANCE','CONCENTRATION','EMISSION'/
      DATA (Pstr(2,I),I=1,4)/'WAVELENGTH','TIME','TEMPERATURE',
     &'DISTANCE'/
      DATA (Pstr(3,I),I=1,11)/'OFF','0.01','0.02','0.05','0.1','0.2',
     &'0.5','1.0','2.0','5.0','10.0'/
      DATA Pstr(4,1)/'OFF'/
      DATA (Pstr(4,I),I=6,15)/'0.2','0.5','1.0','2.0','5.0',
     &'10','20','50','100','200'/
      DATA (Pstr(5,I),I=1,4)/'AUTO SELECT','AUTO GAIN','AUTO SLIT',
     &'SINGLE BEAM'/
      DATA (Pstr(6,I),I=1,5)/'OFF','NORMAL','1ST DERIV','2ND DERIV',
     &'LOG'/
      DATA (Pstr(7,I),I=1,9)/'0.01','0.02','0.05','0.1','0.2','0.5',
     &'1.0','2.0','4.0'/
      DATA (Pstr(8,I),I=10,16)/'2','5','10','20','50','100','200'/
      DATA (Pstr(9,I),I=12,15)/'10','20','50','100'/
      DATA (Pstr(10,I),I=1,5)/'-1.9 TO 0.6','-2.0 TO 0.5',
     &'-2.1 TO 0.4','-2.2 TO 0.3','-2.3 TO 0.2'/
      DATA (Pstr(11,I),I=1,16)/'+/-0.01','+/-0.02','+/-0.05','+/-0.1',
     &'+/-0.2','+/-0.5','+/-1.0','+/-2.0','+/-5.0','+/-10','+/-20',
     &'+/-50','+/-100','+/-200','+/-500','+/-1000'/
      DATA (Pstr(12,I),I=1,16)/'+/-0.01','+/-0.02','+/-0.05','+/-0.1',
     &'+/-0.2','+/-0.5','+/-1.0','+/-2.0','+/-5.0','+/-10','+/-20',
     &'+/-50','+/-100','+/-200','+/-500','+/-1000'/
      DATA (Pstr(13,I),I=1,8)/'+/-0.01','+/-0.02','+/-0.05','+/-0.1',
     &'+/-0.2','+/-0.5','+/-1.0','+/-2.0'/
      DATA (Pstr(14,I),I=1,10)/'0','10','20','30','40','50','60','70',
     &'80','90'/
```

```
      DATA (Pstr(15,I),I=1,4)/'0.5'.'1.0','3.0'.'10'/
      DATA (Pstr(16,I),I=1,2)/'NORMAL','REVERSE'/
      DATA (Pstr(17,I),I=1,2)/'OFF','ON'/
      DATA (Pstr(18,I),I=1,2)/'REPEAT SCAN','SGL/MULTI'/
      DATA (Pstr(19,I),I=1,2)/'SERIAL','OVERLAY'/
      DATA (Pstr(20,I),I=1,4)/'BOTH ON','UV ONLY','VIS/NIR ONLY',
     &'BOTH OFF'/
      DATA (Pstr(21,I),I=1,3)/'AUTO','UV','VIS/NIR'/
      DATA (Pstr(22,I),I=1,3)/'AUTO','UV/VIS','NIR'/
      DATA (Pstr(23,I),I=1,2)/'FULL','1/3'/
      DATA (Pstr(24,I),I=1,3)/'AUTO','UV/VIS','NIR'/
      DATA (Pstr(25,I),I=1,3)/'AUTO','UV','VIS/NIR'/
      DATA (Pstr(26,I),I=1,4)/'AUTO SELECT','AUTO GAIN','AUTO SLIT',
     &'SINGLE BEAM'/
      DATA (Pstr(27,I),I=1,2)/'FULL','1/3'/
      DATA (Pstr(28,I),I=1,6)/'0','1','2','3','4','5'/
      DATA (Pstr(29,I),I=1,2)/'STANDARDS','UNKNOWNS'/
      DATA (Pstr(30,I),I=3,6)/'DIRECT','LINEAR','DIRECT-QUAD',
     &'QUADRATIC'/
      DATA (Pstr(31,I),I=7,8)/'NORMAL','AVERAGED'/
      DATA (Pstr(32,I),I=9,13)/'SIGNAL AV','SAMPLE AV','QUICK',
     &'EXTENDED','FIXED'/
      DATA (Pstr(33,I),I=1,5)/'DSPL RESULTS','DSPL SETUP','NEXT CONC',
     &'DELETE SAMPLE'.'CLEAR RESULTS'/
      DATA (Pstr(34,I),I=1,2)/'OFF','ON'/
      DATA (Pstr(35,I),I=1,2)/'1','2'/
      DATA (Pstr(38,I),I=1,5)/'  OFF','  ON','RECORD',' ','ON/SETUP'/
      DATA (Pstr(40,I),I=1,2)/'INTERVAL','ACCY-DRIVEN'/
C
      DATA (Printer(I),I=1,3)/'WAVELENGTH','TIME','TEMPERATURE'/
      DATA (Printer(I),I=4,6)/'DISTANCE','MAX.mm','MIN.mm'/
C
C         Initialize Screen Control String Variables
C
      BELL=CHAR(7)
      ESC=CHAR(27)
      CLR=ESC//'J'
      HOME=ESC//'h'
      UP=ESC//'A'
      DOWN=ESC//'B'
      ERASE=ESC//'K'
C
C     -----------------------------------------------------------------
C
C         Assign Statement Labels
C
C     -----------------------------------------------------------------
C
      ASSIGN 10  TO MENU
      ASSIGN 90  TO SPECTRUM
      ASSIGN 200 TO BASELINE
      ASSIGN 350 TO ADVANCED
      ASSIGN 390 TO INSTRUMENT
      ASSIGN 400 TO PARAMETERS
      ASSIGN 570 TO LAMP
      ASSIGN 600 TO ACCESSORY
      ASSIGN 700 TO STORE
      ASSIGN 900 TO EXIT
C
C     ----------------------------------------------------------------
C
C         Data Acquisition and Control Menu
C
C     ----------------------------------------------------------------
C
      CALL FFRCL(79)               ! Eliminate Line Wrapping Problems
      CALL Send(Setup)             ! Setup Normal Handshaking With Cary
      CALL Send(Ready)             ! Release Cary From Standby Mode
   10 WRITE (1,*) HOME,CLR
      CALL Send(Instr)             ! Display Instrument Settings On Cary
      CALL Send(Messoff)           ! Turn Off Display Messages On Cary
      CALL Send(Unlock)            ! Unlock Keyboard On Cary
      NCOL=70                      ! Set Display To 70 Columns
      TITLE='Cary 2390'
      CALL Center(TITLE)
      TITLE='Spectral Data Acquisition'
      CALL Center(TITLE)
      WRITE (1,'(T61,A2,A8)') UP,'Rev: 1.7'
      CALL Line(NCOL)
      WRITE (1,20) 'CODE','FUNCTION','STATUS','MIN','MAX','INC'
   20 FORMAT (T4,A4,T14,A8,T34,A7,T50,A3,T58,A3,T66,A3)
      CALL Line(NCOL)
      WRITE (1,30) 'A.....Acquire Spectrum.......',Sstat,Smin,Smax,Sinc
      WRITE (1,40) 'B.....Baseline Setup.........',Bstat,Bmin,Bmax
      WRITE (1,50) 'I.....Instrument Settings....'
      WRITE (1,50) 'L.....Lamps/Detectors/Access.'
      WRITE (1,60) 'S.....Store File on Disk.....',Fname
      WRITE (1,50) 'X.....EXIT Data Acquisition..'
   30 FORMAT (/,T4,A30,T35,A8,T50,A4,T58,A4,T66,A4)
   40 FORMAT (/,T4,A30,T35,A8,T50,A4,T58,A4)
   50 FORMAT (/,T4,A30)
   60 FORMAT (/,T4,A30,T35,A20)
      WRITE (1,*)
      CALL Line(NCOL)
      WRITE (1,*)
   70 WRITE (1,*) UP,ERASE,'_'
      WRITE (1,'(T3,A15,A,A2)') 'Enter the CODE:',BELL,'_'
      READ (1,80) Code
   80 FORMAT (A1)
      CALL Upper(Code)
      IF (Code.EQ.'A') GO TO SPECTRUM
      IF (Code.EQ.'B') GO TO BASELINE
      IF (Code.EQ.'I') GO TO INSTRUMENT
      IF (Code.EQ.'L') GO TO ADVANCED
      IF (Code.EQ.'S') GO TO STORE
      IF (Code.EQ.'X') GO TO EXIT
      GO TO 70
C
```

```
C     --------------------------------------------------------------
C
C         Acquire Spectrum: (Instrument Baseline Must Match)
C
C     --------------------------------------------------------------
C
   90 CALL Partable(PARAM)
      IF ((PARAM(38).NE.1).OR.(Bstat.EQ.'  OFF')) THEN
        WRITE (1,*) UP,ERASE,'_'
        WRITE (1,*) ' Baseline Program Is ABSENT: ',BELL,'_'
        CALL Wait(2.0)
        WRITE (1,*)
        GO TO BASELINE
      END IF
      IF ((PARAM(1).EQ.2).OR.(PARAM(2).NE.0)) THEN
        WRITE (1,*) UP,ERASE,'_'
        WRITE (1,*) ' Ordinate or Abscissa Error: ',BELL,'_'
        CALL Wait(2.0)
        WRITE (1,*)
        GO TO INSTRUMENT
      END IF
      IF (Sstat.EQ.'ACQUIRED') THEN
  100   WRITE (1,*) UP,ERASE,'  SPECTRUM NOT STORED:_'
        WRITE (1,*) ' Proceed With Spectrum (Y or N) ? ',BELL,'_'
        READ (1,80) Code
        CALL Upper(Code)
        IF (Code.EQ.'N') GO TO 70
        IF (Code.NE.'Y') GO TO 100
      END IF
C
C     --------------------------------------------------------------
C
C         Store Previous Spectrum's Parameters For Possible Abort
C
C     --------------------------------------------------------------
C
      Omin=Smin
      Omax=Smax
      Oinc=Sinc
      Odet=Sdet
      Ogain=Sgain
      Olamp=Slamp
      Operiod=Speriod
      Orate=Srate
      Oref=Sref
      Oslit=Sslit
      Osbw=Ssbw
C     --------------------------------------------------------------
      WRITE (1,*) HOME,CLR
      TITLE='Scan Parameters'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,*) DOWN,'  BASELINE:'
      WRITE (1,*) DOWN,'  Scan Limits, (nm): ',Bmin,'/ ',Bmax
      CALL Val(Bmin,WMIN)                 ! Default Spectrum To
      CALL Val(Bmax,WMAX)                 ! Baseline Scan Limits
      WRITE (1,*) DOWN,DOWN,'  SPECTRUM:'
      WRITE (1,*) DOWN,'  Scan Limits, (nm): ',Bmin,'/ ',Bmax,
     &DOWN,DOWN
  110 WRITE (1,*) UP,ERASE,'  A...Accept, C...Change, X...Exit ? ',
     &BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'X') GO TO MENU
      IF (Code.EQ.'A') GO TO 120
      IF (Code.NE.'C') GO TO 110
      CALL Limits(WMIN,WMAX)
  120 CALL Str(WMIN,String,4)
      Smin=String(2:5)
      CALL Str(WMAX,String,4)
      Smax=String(2:5)
      WRITE (1,*) UP,ERASE,UP,UP,ERASE,'  Scan Limits, (nm): ',
     &Smin,'/ ',Smax,DOWN,DOWN
  130 WRITE (1,*) UP,ERASE,'  Step Size (.01 - 5 nm) : ',BELL,'_'
      READ (1,*,ERR=130) STEP
      IF ((STEP.LT.0.01).OR.(STEP.GT.5.0)) GO TO 130
      CALL Str(STEP,String,4)
      Sinc=String(2:5)
  140 CALL Val(Pstr(3,PARAM(3)+1),RATE)
      RATE=RATE/STEP
      IF (RATE.GT.5.0) THEN
        WRITE (1,*) UP,ERASE,'  Data Rate > 5 Hz - _'
        WRITE (1,*) 'RESET Scan Rate, (Y or N) ? ',
     &  BELL,'_'
        READ (1,80) Code
        CALL Upper(Code)
        IF (Code.NE.'Y') GO TO 130
        N=3
        K=N
        CALL Select(N,PARAM,Pstr)
        PARAM(K)=N-1                      ! Update Parameter Table
        GO TO 140
      END IF
      NDATA=INT((WMAX-WMIN)/STEP+.5)+1
      IF (NDATA.GT.10001) THEN
        WRITE (1,*) UP,ERASE,'  Too Many Data Points - _'
        WRITE (1,*) 'Increase Step Size _',BELL
        CALL Wait(2.0)
        GO TO 130
      END IF
      WRITE (1,*) DOWN,'  Checking Instrument Settings:',BELL
      CALL GOTO(Bmax)        ! Test Matching At Start Of Baseline Scan
C
C     ----------------------------------------------------------------
C
C         Set Spectrum Strings to Match Instrument Parameters
C
C     ----------------------------------------------------------------
```

```
C
      Sdet=CHAR(PARAM(22)+48)
      Slamp=CHAR(PARAM(21)+48)
      Speriod=CHAR(PARAM(15)+48)
      Srate=CHAR(PARAM(3)+48)
      Sref=CHAR(PARAM(5)+48)
      Sslit=CHAR(PARAM(23)+48)
      CALL Vartable(VARIABLE)               ! Update SBW, GAIN at Bmax
      CALL Str(VARIABLE(10),String,4)
      Ssbw=String(2:5)
      CALL Str(VARIABLE(6),String,4)
      Sgain=String(2:5)
C
C     ----------------------------------------------------------------
C
C       Test For Acceptable Instrument Baseline Matching
C
C     ----------------------------------------------------------------
C
      MATCH=.TRUE.
      IF (WMAX.GT.VARIABLE(3)) MATCH=.FALSE.
      IF (WMIN.LT.VARIABLE(4)) MATCH=.FALSE.
      IF (Sref.NE.Bref) MATCH=.FALSE.
      IF (Slamp.NE.Blamp) MATCH=.FALSE.
      IF (Sdet.NE.Bdet) MATCH=.FALSE.
      IF (Sslit.NE.Bslit) MATCH=.FALSE.
C     ----------------------------------------------------------------
      IF ((MODE.EQ.1).AND.(MATCH)) THEN     ! Exit If Already Failed
        IF (Ssbw.NE.Bsbw) THEN
          WRITE (1,*) UP,ERASE,'  Matching To Baseline SBW:',BELL
          Command=Varset//'9'//Bsbw//'!0'
          CALL Send(Command)
          Ssbw=Bsbw
          CALL Wait(1.0)
        END IF
      END IF
C     ----------------------------------------------------------------
      IF ((MODE.EQ.2).AND.(MATCH)) THEN     ! Exit If Already Failed
        IF (Sgain.NE.Bgain) THEN
          WRITE (1,*) UP,ERASE,'  Matching To Baseline GAIN:',BELL
          Command=Varset//'5'//Bgain//'!0'
          CALL Send(Command)                ! Reset AUTOSLIT Gain Level
          Sgain=Bgain
          CALL Wait(2.0)
        END IF
        IF ((Bref.EQ.'0').AND.(Bdet.EQ.'0').AND.(WMAX.LE.800.0)) THEN
          WRITE (1,*) UP,ERASE,'  Matching To Baseline SBW:',BELL
          Wlength='800.5'
          CALL GOTO(Wlength)                ! Reset To NIR Region
          CALL Wait(1.0)
          Wlength='800.0'
          CALL GOTO(Wlength)                ! Set To Start Of UV/VIS
        END IF                              ! With Matching SBW
      END IF
      IF (MATCH) GO TO 150
C
C     ----------------------------------------------------------------
C
C         Record New Baseline Using Present Instrument Parameters
C         With Period & Scan Rate From The Previous Baseline Scan
C
C     ----------------------------------------------------------------
C
      WRITE (1,*) DOWN,DOWN
      TITLE='### NEW BASELINE REQUIRED ###'
      CALL Center(TITLE)
      WRITE (1,*) BELL
      CALL Wait(2.0)
      MATCH=.TRUE.                        ! Baseline Valid Test On Exit
      CALL Bline(WMIN,WMAX,Sdet,Sgain,Slamp,Fperiod,Frate,Sref,Ssbw,
     &Sslit,Bscan,Btime,MATCH,MODE)
      Command=Parset//'>'//Speriod//CSM   ! Reset To Spectrum's Period
      CALL Send(Command)
      Command=Parset//'2'//Srate//CSM     ! Reset To Spectrum's Rate
      CALL Send(Command)
      IF (.NOT.MATCH) THEN
        Sstat='  OFF'                     ! Aborted Scan Exit
        GO TO MENU
      END IF
C
C     ----------------------------------------------------------------
C
C         Update Baseline Parameter Strings
C
C     ----------------------------------------------------------------
C
      Bmin=Smin
      Bmax=Smax
      Bdet=Sdet
      Bgain=Sgain
      Blamp=Slamp
      Bperiod=Fperiod
      Brate=Frate
      Bref=Sref
      Bsbw=Ssbw
      Bslit=Sslit
      Bstat='ON/MATCH'
C     ----------------------------------------------------------------
  150 WRITE (1,*) HOME,CLR
      TITLE='Acquire Spectrum'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,*) DOWN,'  Wavelength Limits, (nm): ',Smax,'/ ',Smin
      WRITE (1,*) DOWN,'  Step Size, (nm/datum)  : ',Sinc
      WRITE (1,*) DOWN,'  Scan Rate, (nm/sec)    : ',
     &Pstr(3,PARAM(3)+1)
      WRITE (1,*) DOWN,'  Response Time, (sec)   : ',
     &Pstr(15,PARAM(15)+1)
```

```
      IF (MODE.EQ.1) THEN
        WRITE (1,*) DOWN,'  Spectral Bandwidth,(nm): ',Ssbw
        GO TO 160
      END IF
      WRITE (1,*) DOWN,'  AUTOSLIT Gain Level     : ',Sgain
  160 WRITE (1,*) DOWN
      WRITE (1,*) DOWN,'  Place Solution Cell In The SAMPLE Beam:'
      WRITE (1,*) DOWN,'      S.....Start Scan'
      WRITE (1,*) DOWN,'      A.....Abort Scan'
      WRITE (1,*) DOWN,'  Enter the CODE: ',BELL,'_'
  170 READ (1,80) Code
      CALL Upper(Code)
C
C     ---------------------------------------------------------------
C
C         Restore Old Spectrum's Parameter Strings
C
C     ---------------------------------------------------------------
C
      IF (Code.EQ.'A') THEN
        Smin=Omin
        Smax=Omax
        Sinc=Oinc
        Sdet=Odet
        Sgain=Ogain
        Slamp=Olamp
        Speriod=Operiod
        Srate=Orate
        Sref=Oref
        Sslit=Oslit
        Ssbw=Osbw
        GO TO MENU
      END IF
      IF (Code.NE.'S') GO TO 170
C
C     ---------------------------------------------------------------
C
C         Set To Starting Wavelength - Check For Instrument Ready
C
C     ---------------------------------------------------------------
C
      WRITE (1,*) UP,ERASE,UP,UP,ERASE,UP,UP,ERASE,UP,UP,ERASE,'_'
      WRITE (1,*) ' Slewing to Starting Wavelength:',BELL
      CALL GOTO(Smax)
      SINGLE=.FALSE.           ! Scan Mode ON, Single Wavelength OFF
      PRINT=.FALSE.            ! Initialize Print Mode To OFF
  180 WRITE (1,*) UP,ERASE,'  Print to Screen, (Y or N) ? ',BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'Y') THEN
        PRINT=.TRUE.
        GO TO 190
      END IF
      IF (Code.NE.'N') GO TO 180
```

```
C
C     ----------------------------------------------------------------
C
C         Select Data String Format For Abscissa & Ordinate In
C         SUBROUTINE Acquire Via COMMON Variables XMODE & YMODE
C
C     ----------------------------------------------------------------
C
  190 YMODE=PARAM(1)+1          ! YMODE = 1 - 6 (Only 1 & 2 Allowed)
      XMODE=PARAM(2)+1          ! XMODE = 1 - 4 (Only 1 Allowed)
      SPECGAIN=VARIABLE(6)      ! Save GAIN Value At Smax For Data File
C
      WRITE (1,*) UP,ERASE,'  Scanning Spectrum:',BELL
C
      CALL Send(Instr)          ! Display Instrument Settings On Cary
      CALL Send(Lock)           ! Lock Keyboard On Cary During Scan
      CALL Wait(1.0)            ! Wait For Cary To Finish Housekeeping
C
C     ----------------------------------------------------------------
C                   *** Data Collection Subroutine ***
C
C         Collects NDATA Readings At Sinc (nm) Steps And
C         Returns Spectrum In Arrays (X),(Y) Via EMA COMMON
C         Final Reading Is Returned Via Named COMMON /CARY/
C
      CALL Acquire(Sinc,PRINT,SINGLE,WAVELENGTH)
C
C     ----------------------------------------------------------------
C
      CALL Terminate            ! UNTALK Cary 2390 From IEEE-488 Bus
      CALL Wait(1.0)            ! Wait For Cary To Finish Housekeeping
      CALL Send(Setup)          ! Re-establish Normal Handshaking
      CALL Send(Stop)           ! STOP Key Issued
      CALL Send(Unlock)         ! UNLOCK Keyboard
      CALL Vartable(VARIABLE)   ! Update Instrument Variables To Obtain
      SPECBAND=VARIABLE(10)     ! Value Of Spectral Bandwidth At Smin.
      CALL GOTO(Smax)           ! Return To Starting Wavelength
      NARRAY=NDATA              ! Save # Of Data Points In File Variable
      XMIN=WMIN                 ! Save End Of Scan In File Variable
      XMAX=WMAX                 ! Save Start Of Scan In File Variable
      XSTEP=STEP                ! Save Step Size In File Variable
      Sstat='ACQUIRED'
      GO TO MENU
C
C     ----------------------------------------------------------------
C
C         Baseline Call and Status Check
C
C     ----------------------------------------------------------------
C
```

```
  200 MATCH-.TRUE.
      TRANSFER-.FALSE.
  210 WRITE (1,*) UP,ERASE,'  Reading Instrument Baseline: ',BELL,'_'
      CALL Partable(PARAM)
      Bstat-Pstr(38,PARAM(38)+1)
      IF (PARAM(38).GT.1) Bstat-'  '//Pstr(38,PARAM(38)+1)
      CALL Vartable(VARIABLE)
      WMAX-NINT(VARIABLE(3))
      WMIN-NINT(VARIABLE(4))
      BAND-VARIABLE(2)/1000.0      ! Only One Of SBW Or GAIN Is Stored
      GAIN-VARIABLE(2)/10.0        ! By The Cary For The Baseline Scan
  220 PREC-4                       ! - Decide Below Which Is Valid.
      IF (WMAX.LT.1000.0) PREC-3
      CALL Str(WMAX,String,PREC)
      Bmax-String(2:5)
      PREC-4
      IF (WMIN.LT.1000.0) PREC-3
      CALL Str(WMIN,String,PREC)
      Bmin-String(2:5)
C     ----------------------------------------------------------------
      IF (TRANSFER) GO TO MENU     ! EXIT After Return From Bline
C     ----------------------------------------------------------------
      IF (WMAX.GT.900.0) THEN
        IF (PARAM(24).EQ.1) PARAM(24)-0    ! Bad UV/VIS Detector Mode
        IF (PARAM(26).EQ.1) PARAM(26)-0    ! Bad AUTO GAIN Ref. Mode
      END IF
      IF (WMIN.LT.700.0) THEN
        IF (PARAM(24).EQ.2) PARAM(24)-0    ! Bad NIR Detector Mode
      END IF
      IF (PARAM(24).EQ.2) THEN
        IF (PARAM(26).EQ.1) PARAM(26)-0    ! Bad NIR Reference Mode
      END IF
      IF ((PARAM(24).EQ.0).AND.(PARAM(26).EQ.1)) THEN
        IF (WMAX.GT.800.0) PARAM(24)-1     ! Bad AUTO Detector Mode
      END IF
      IF (WMAX.GT.400.0) THEN
        IF (PARAM(25).EQ.1) PARAM(25)-0    ! Bad UV Lamp Mode
      END IF
      IF (WMIN.LT.270.0) THEN
        IF (PARAM(25).EQ.2) PARAM(25)-0    ! Bad W Lamp Mode
      END IF
C     ----------------------------------------------------------------
      Bperiod-CHAR(PARAM(15)+48)
      Brate-CHAR(PARAM(3)+48)
      Bdet-CHAR(PARAM(24)+48)
      Blamp-CHAR(PARAM(25)+48)
      Bref-CHAR(PARAM(26)+48)
      Bslit-CHAR(PARAM(27)+48)
      IF (Bref.EQ.'2') GO TO 230    ! AUTOSLIT Mode On (Both Detectors)
      IF (Bdet.EQ.'2') GO TO 230    ! NIR Detector  -> AUTOSLIT Mode
      IF (WMAX.GT.900.0) GO TO 230 ! Lambda > 900  -> AUTOSLIT Mode
      IF (WMAX.GT.800.0) THEN
        IF (Bdet.EQ.'0') GO TO 230 ! AUTO Detector -> AUTOSLIT Mode
      END IF
```

```
C     ----------------------------------------------------------------
      CALL Str(BAND,String,4)
      Bsbw=String(2:5)             ! SBW Fixed At The Start Of Scan
      Bgain=' '                    ! Gain Variable During Scan
      MODE=1
      GO TO 240
  230 CALL Str(GAIN,String,4)
      Bgain=String(2:5)            ! GAIN Fixed At The Start Of Scan
      Bsbw=' '                     ! SBW Variable During Scan
      MODE=2
  240 Command=Key//'H0'            ! Display Baseline Menu On Cary
      CALL Send(Command)
      CALL Send(Messoff)           ! Turn Off Any Display Messages
C     ----------------------------------------------------------------
      WRITE (1,*) HOME,CLR
      TITLE='Baseline Setup'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,250) 'INDEX','FUNCTION','SETTING'
      CALL LINE(NCOL)
      WRITE (1,*)
      WRITE (1,260) '0:','.....AUTO BALANCE...........',
     &Astat
      WRITE (1,260) '1:','.....BASELINE STATUS........',
     &Bstat
      WRITE (1,280) '2:','.....WAVELENGTH (Max,Min)...',
     &Bmax,',',Bmin
      WRITE (1,280) '3:','.....SBW (nm), GAIN.........',
     &Bsbw,',',Bgain
      WRITE (1,270) '4:','.....REFERENCE MODE.........',
     &Pstr(26,PARAM(26)+1)
      WRITE (1,270) '5:','.....LAMP SELECT............',
     &Pstr(25,PARAM(25)+1)
      WRITE (1,270) '6:','.....DETECTOR SELECT........',
     &Pstr(24,PARAM(24)+1)
      WRITE (1,270) '7:','.....SLIT HEIGHT............',
     &Pstr(27,PARAM(27)+1)
      WRITE (1,270) '8:','.....SCAN RATE (nm/sec).....',
     &Pstr(3,PARAM(3)+1)
      WRITE (1,270) '9:','.....RESPONSE TIME (sec)....',
     &Pstr(15,PARAM(15)+1)
      WRITE (1,260) 'X:','.....EXIT Baseline Menu.....',
     &' '
  250 FORMAT (T4,A5,T20,A8,T40,A7)
  260 FORMAT (T4,A3,T10,A28,A8)
  270 FORMAT (T4,A3,T10,A28,T40,A14)
  280 FORMAT (T4,A3,T10,A28,T40,A4,A,A4)
      WRITE (1,*)
      CALL Line(NCOL)
      WRITE (1,*)
  290 WRITE (1,*) UP,ERASE,'  INDEX Code: ',BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'X') GO TO 330
      N=ICHAR(Code)-48
      IF ((N.LT.0).OR.(N.GT.9)) GO TO 290
C     -----------------------------------------------------------
      IF (N.EQ.0) THEN
        CALL Send(Autobal)
        Astat='  SET'
        GO TO 220
      END IF
      IF (N.EQ.1) N=38
      IF (N.EQ.2) GO TO 300        ! Update Wavelength Limits
      IF (N.EQ.3) GO TO 310        ! Update SBW/GAIN
      IF (N.EQ.4) N=26
      IF (N.EQ.5) N=25
      IF (N.EQ.6) N=24
      IF (N.EQ.7) N=27             ! Slit Control Is Manual Only
      IF (N.EQ.8) N=3
      IF (N.EQ.9) N=15
C     -----------------------------------------------------------
      K=N                          ! Instrument Baseline PARAMETERS Are
      CALL Select(N,PARAM,Pstr)    ! Masked From Direct Changes - The
      PARAM(K)=N-1                 ! NEW Values Are Only Accepted From
      IF (K.EQ.38) THEN            ! SUBROUTINE Bline's '@J' Command.
        WRITE (1,'(T12,A)') '_'
        GO TO 200
      END IF
      GO TO 220
C     -----------------------------------------------------------
  300 CALL Limits(WMIN,WMAX)
      MATCH=.FALSE.
      GO TO 220
  310 WRITE (1,*) UP,ERASE,'_'
      IF (MODE.EQ.2) GO TO 320
      WRITE (1,*) ' Spectral Bandwidth (0.04 - 3.60 nm) = ',BELL,'_'
      READ (1,*,ERR=310) BAND
      IF ((BAND.LT.0.04).OR.(BAND.GT.3.60)) GO TO 310
      GO TO 220
  320 String='(1-1275)'
      IF (PARAM(24).NE.2) String='(1 - 1000)'
      WRITE (1,*) ' Gain Level ',String(1:10),' = ',BELL,'_'
      READ (1,*,ERR=310) NUMBER
      IF ((NUMBER.LT.1.0).OR.(NUMBER.GT.1275.0)) GO TO 310
      IF ((NUMBER.GT.1000.0).AND.(PARAM(24).NE.2)) GO TO 310
      GAIN=VARIABLE(6)
      RATIO=NUMBER/GAIN
      IF (RATIO.GT.10.0) THEN
        WRITE (1,*) UP,ERASE,'_'
        WRITE (1,*) ' Setting Instrument Gain: ',BELL,'_'
        CALL Wait(2.0)
        Command=Parset//'420'          ! Set AUTOSLIT Mode Prior
        CALL Send(Command)             ! To Sending New GAIN Level
        WRITE (1,*)
        TRANSFER=.TRUE.                ! Transfer To Instrument
        GO TO 490                      ! GAIN Setting Routine
      END IF
      GAIN=NUMBER
      GO TO 220
C
C     ----------------------------------------------------------------
C
C         Record Baseline Scan In CARY 2390
C
C     ----------------------------------------------------------------
C
  330 WRITE (1,*) UP,ERASE,'_'
      WRITE (1,*) ' Record NEW Baseline, (Y or N) ? ',BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'N') GO TO 340
      IF (Code.NE.'Y') GO TO 330
      MATCH=.FALSE.
      I=ICHAR(Brate)-48
      Bscan=Pstr(3,I+1)
      I=ICHAR(Bperiod)-48
      Btime=Pstr(15,I+1)
      CALL Bline(WMIN,WMAX,Bdet,Bgain,Blamp,Bperiod,Brate,Bref,Bsbw,
     &Bslit,Bscan,Btime,MATCH,MODE)
      Fperiod=Bperiod
      Frate=Brate
      Astat='  SET'
  340 TRANSFER=.TRUE. ! Perform An Alternate Return To The Main MENU
      GO TO 210       ! After Reading Instrument Baseline Parameters
C
C     ---------------------------------------------------------- ------
C
C         Menu of Advanced Setup Operations
C
C     ----------------------------------------------------------------
C
  350 WRITE (1,*) HOME,CLR
      TITLE='Advanced Operations Menus'
      CALL Center(TITLE)
      NCOL=50
      CALL Line(NCOL)
      WRITE (1,'(T15,A5,T30,A14)') 'INDEX','GROUP FUNCTION'
      CALL Line(NCOL)
      WRITE (1,*)
      WRITE (1,360) '1: .......LAMPS & DETECTORS........'
      WRITE (1,360) '2: .......ACCESSORY SETTINGS.......'
      WRITE (1,360) '3: .......AUTOMATIC OPERATION......'
      WRITE (1,360) 'X: .......EXIT TO SETUP MENU.......'
      WRITE (1,*)
      CALL Line(NCOL)
  360 FORMAT (T17,A35)
      WRITE (1,*)
  370 WRITE (1,'(T15,A2,A2,A9,A,A)') UP,ERASE,'INDEX #: ',BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'X') GO TO MENU
      N=ICHAR(Code)-48
      IF (N.EQ.1) GO TO LAMP
      IF (N.EQ.2) GO TO ACCESSORY
      IF (N.EQ.3) GO TO 380
      GO TO 370
  380 WRITE (1,'(T13,A2,A2,A)') UP,ERASE,'_'
      WRITE (1,*) ' Not Supported In Version 1.X ',BELL,'_'
      CALL Wait(2.0)
      GO TO ADVANCED
C
C     ---------------------------------------------------------------
C
C         Display and Update Instrument Settings
C
C     ---------------------------------------------------------------
C
  390 WRITE (1,*) UP,ERASE,'  Reading Wavelength: ',BELL,'_'
      NDATA=1
      Oinc='1'
      PRINT=.FALSE.                     ! No Display Required
      SINGLE=.TRUE.                     ! Select Wavelength Update Mode
      CALL Partable(PARAM)
      YMODE=PARAM(1)+1                  ! Set Data String Format
      XMODE=2
      Icode=CHAR(PARAM(2)+48)           ! Save Abscissa Mode
      Command=Parset//'110'             ! Set Abscissa = TIME
      CALL Send(Command)
      CALL Send(Setup)
      CALL Wait(0.5)
      CALL Acquire(Oinc,PRINT,SINGLE,WAVELENGTH)
      CALL Terminate
      CALL Wait(0.5)
      CALL Send(Setup)
      CALL Send(Stop)
      Command=Parset//'1'//Icode//CSM
      CALL Send(Command)                ! Restore Abscissa Mode
      WRITE (1,*)
      WRITE (1,*) UP,'_'
  400 WRITE (1,*) ERASE,' Reading Instrument Settings: ',BELL,'_'
      CALL Partable(PARAM)
      CALL Vartable(VARIABLE)
      BAND=VARIABLE(10)                     ! Current SBW (nm)
      GAIN=VARIABLE(6)                      ! Current GAIN Level
      CALL Val(Pstr(8,PARAM(8)+1),PMAX)     ! Pen Limits, %T & %R
      PMIN=VARIABLE(11)
      IF (PARAM(1).EQ.0) THEN               ! Pen Limits, Absorbance
        CALL Val(Pstr(7,PARAM(7)+1),PMAX)
        PMIN=VARIABLE(1)
      END IF
      IF (PARAM(1).EQ.2) THEN               ! Pen Limits, Temperature
        CALL Val(Pstr(9,PARAM(9)+1),PMAX)
        CALL Val(Pstr(14,PARAM(14)+1),PMIN)
      END IF
      PMAX=PMIN+PMAX
      I=11                              ! Index For Deriv. Range
      IF (PARAM(6).EQ.4) I=10           ! Index For Log Zero Range
      String=Pstr(I,PARAM(I)+1)         ! Pen Range Label For Index
      CALL Send(Instr)                  ! Display Instrument Menu
C     -----------------------------------------------------------------
      WRITE (1,*) HOME,CLR
      TITLE='Instrument Settings'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,'(T4,A5,T20,A8,T40,A7)') 'INDEX','FUNCTION','SETTING'
      CALL LINE(NCOL)
      WRITE (1,*)
      WRITE (1,410) '0:','.....WAVELENGTH.............',
     &WAVELENGTH
      WRITE (1,420) '1:','.....ORDINATE...............',
     &Pstr(1,PARAM(1)+1)
      WRITE (1,420) '2:','.....ABSCISSA...............',
     &Pstr(2,PARAM(2)+1)
      WRITE (1,420) '3:','.....SCAN RATE (nm/sec).....',
     &Pstr(3,PARAM(3)+1)
      WRITE (1,420) '4:','.....CHART DISPLAY (nm/cm)..',
     &Pstr(4,PARAM(4)+1)
      WRITE (1,420) '5:','.....REFERENCE MODE.........',
     &Pstr(5,PARAM(5)+1)
      WRITE (1,430) '6:','.....SBW (nm), GAIN.........',
     &BAND,',',GAIN
      WRITE (1,420) '7:','.....PEN FUNCTION...........',
     &Pstr(6,PARAM(6)+1)
      WRITE (1,430) '8:','.....PEN LIMITS (Min,Max)...',
     &PMIN,',',PMAX
      IF (PARAM(6).GT.1) WRITE (1,'(T40,A2,A2,A14)') UP,ERASE,String
      WRITE (1,420) '9:','.....RESPONSE TIME (sec)....',
     &Pstr(15,PARAM(15)+1)
      WRITE (1,420) '10:','.....BEAM INTERCHANGE.......',
     &Pstr(16,PARAM(16)+1)
      WRITE (1,420) '11:','.....SLIT HEIGHT............',
     &Pstr(23,PARAM(23)+1)
      WRITE (1,420) 'X:','.....EXIT Instrument Menu...',' '
  410 FORMAT (T4,A3,T10,A28,T40,F6.2)
  420 FORMAT (T4,A3,T10,A28,T40,A14)
  430 FORMAT (T4,A3,T10,A28,T40,F4.2,A,F5.2)
      WRITE (1,*)
      CALL Line(NCOL)
      WRITE (1,*)
  440 WRITE (1,*) UP,ERASE,'  INDEX Code: ',BELL,'_'
      READ (1,'(A2)') Key
      CALL Upper(Key)
      IF (Key.EQ.'X') GO TO MENU
      N=ICHAR(Key(1:1))-48
      IF (Key(2:2).EQ.' ') GO TO 450
      N=N*10+ICHAR(Key(2:2))-48
  450 IF ((N.LT.0).OR.(N.GT.10)) GO TO 440
C     -----------------------------------------------------------------
```

```
      K=N                                  ! Save Index #:
      IF (N.EQ.0) GO TO 460                ! Update Wavelength
      IF (N.EQ.6) GO TO 470                ! Update SBW/GAIN
      IF (N.EQ.7) N=6                      ! Pen Function
      IF (N.EQ.9) N=15
      IF (N.EQ.10) N=16
      IF (N.EQ.8) THEN                     ! Update Pen Limits
        N=1                                ! Index For Deriv & Log
        IF (PARAM(6).LE.1) THEN            ! PEN = NORMAL Modes
          IF (PARAM(1).NE.2) GO TO 520     ! Absorbance & %T Range
          N=9                              ! Index For Temp. Range
        END IF
      END IF
      CALL Select(N,PARAM,Pstr)            ! Update Parameters
      IF (K.EQ.5) THEN
        WRITE (1,'(T12,A)') '_'
        GO TO 510
      END IF
      WRITE (1,'(T13,A2,A)') UP,'_'
      GO TO PARAMETERS
C     ------------------------------------------------------------
  460 WRITE (1,*) UP,ERASE,'  Wavelength = ',BELL,'_'
      READ (1,*,ERR=460) NUMBER
      IF ((NUMBER.LT.185.0).OR.(NUMBER.GT.3152)) GO TO 460
      IF ((Bdet.EQ.'1').AND.(NUMBER.GT.900.0)) THEN
        WRITE (1,*) UP,ERASE,'  UV/VIS Detector Limit = 900 nm',BELL
        CALL Wait(2.0)
        GO TO 460
      END IF
      IF ((Bdet.EQ.'2').AND.(NUMBER.LT.700.0)) THEN
        WRITE (1,*) UP,ERASE,'  NIR Detector Limit = 700 nm',BELL
        CALL Wait(2.0)
        GO TO 460
      END IF
      CALL Str(NUMBER,String,6)
      Wlength=String(2:8)
      WRITE (1,*) UP,ERASE,'  Slewing to _'
      WRITE (1,'(F6.2,A4)') NUMBER,' nm:'
      CALL GOTO(Wlength)
      GO TO INSTRUMENT
C     ------------------------------------------------------------
  470 WRITE (1,*) UP,ERASE,'_'
      IF (PARAM(5).EQ.2) GO TO 480     ! AUTOSLIT Mode (Both Detectors)
      IF (PARAM(22).EQ.2) GO TO 480    ! NIR Detector  => AUTOSLIT Mode
      IF (WAVELENGTH.GT.900.0) THEN
        GO TO 480                      ! Lamda >900 nm => AUTOSLIT Mode
      END IF
      IF (WAVELENGTH.GT.800.0) THEN
        IF (PARAM(22).EQ.0) GO TO 480 ! AUTO Detector => AUTOSLIT Mode
      END IF
      WRITE (1,*) ' Spectral Bandwidth: (0.04 - 3.60 nm) = ',BELL,'_'
      READ (1,*,ERR=470) BAND
      IF ((BAND.LT.0.04).OR.(BAND.GT.3.60)) GO TO 470
      CALL Str(BAND,String,4)
      Command=Varset//'9'//String(2:5)//'!0'
      CALL Send(Command)
      GO TO 510
  480 String='(1 - 1275)'
      IF (PARAM(22).NE.2) String='(1 - 1000)'
      WRITE (1,*) ' GAIN: ',String(1:10),' = ',BELL,'_'
      READ (1,*,ERR=470) NUMBER
      IF ((NUMBER.LT.1.0).OR.(NUMBER.GT.1275.0)) GO TO 470
      IF ((NUMBER.GT.1000.0).AND.(PARAM(22).NE.2)) GO TO 470
      TRANSFER=.FALSE.
C     ********************** SPECIAL ENTRY POINT *********************
C         Baseline GAIN Request > 10*GAIN : Reset GAIN and RETURN
C     ****************************************************************
  490 J=0
      RATIO=NUMBER/GAIN
      DO WHILE (RATIO.GT.10.0)
        J=J+1
        NUMBER=NUMBER/10.0
        RATIO=NUMBER/GAIN
      END DO
      CALL Str(NUMBER,String,4)
      Command=Varset//'5'//String(2:5)//'!0'
      CALL Send(Command)
      DO 500 I=1,J
        NUMBER=NUMBER*10.0
        CALL Str(NUMBER,String,4)
        CALL Wait(2.0)
        Command=Varset//'5'//String(2:5)//'!0'
        CALL Send(Command)
  500 CONTINUE
      IF (TRANSFER) THEN         ! Return to Baseline Setup
        GAIN=NUMBER              ! With Instrument GAIN Matched
        TRANSFER=.FALSE.         ! To Requested Baseline Gain
        GO TO 220
      END IF
C     ----------------------------------------------------------------
  510 WRITE (1,*) UP,ERASE,'  Waiting for CARY to settle: ',BELL,'_'
      CALL Wait(5.0)
      WRITE (1,*)
      IF (K.EQ.5) WRITE (1,'(T12,A)') '_'
      GO TO 560
C     ----------------------------------------------------------------
  520 N=8
      IF (PARAM(1).EQ.0) N=7
      I=N
      CALL Select(I,PARAM,Pstr)
      I=I+OFFSET(N)
      CALL Val(Pstr(N,I),NUMBER)
      ZERO=0.0
      IF (N.EQ.8) THEN
        IF (NUMBER.GT.100.0) GO TO 550
        GO TO 530
      END IF
      IF (NUMBER.GE.1.0) THEN
        NUMBER=4.0-NUMBER
        GO TO 530
      END IF
      NUMBER=3.0
  530 WRITE (1,'(T13,A2,A2,A18)') UP,ERASE,'  Zero Suppress: _'
      IF (N.EQ.8) THEN
        WRITE (1,*) '(0 - 100%) = ',BELL,'_'
        GO TO 540
      END IF
      WRITE (1,'(A9,F4.2,A4,A,A)') '(-0.5 to ',NUMBER,') = ',BELL,'_'
  540 READ (1,*,ERR=530) ZERO
      IF (N.EQ.8) THEN
        IF ((ZERO.LT.0.0).OR.(ZERO.GT.100.0)) GO TO 530
        GO TO 550
      END IF
      IF ((ZERO.LT.-0.5).OR.(ZERO.GT.NUMBER)) GO TO 530
  550 CALL Str(ZERO,String,3)
      Pcode='0'
      IF (N.EQ.8) Pcode=':'
      Command=Varset//Pcode//String(1:5)//'!0'
      CALL Send(Command)
      WRITE (1,'(T12,A)') '_'
  560 WRITE (1,*) UP,'_'
      GO TO PARAMETERS
C
C     ---------------------------------------------------------------
C
C         Lamp and Detector Mode Selection
C
C     ---------------------------------------------------------------
C
  570 CALL Partable(PARAM)
      WRITE (1,*) HOME,CLR
      TITLE='Lamp & Detector Modes'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,'(T15,A5,T27,A8,T47,A4)') 'INDEX','FUNCTION','MODE'
      CALL Line(NCOL)
      WRITE (1,*)
      WRITE (1,580) '1: .....LAMP POWER.......... ',
     &Pstr(20,PARAM(20)+1)
      WRITE (1,580) '2: .....LAMP SELECT......... ',
     &Pstr(21,PARAM(21)+1)
      WRITE (1,580) '3: .....DETECTOR SELECT..... ',
     &Pstr(22,PARAM(22)+1)
      WRITE (1,580) 'X: .....EXIT TO MENU........ ',' '
      WRITE (1,*)
      CALL LINE(NCOL)
  580 FORMAT (T16,A30,A14)
      WRITE (1,*)
  590 WRITE (1,'(T15,A2,A2,A9,A,A)') UP,ERASE,'INDEX #: ',BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'X') GO TO ADVANCED
      N=ICHAR(Code)-48
      IF ((N.LT.1).OR.(N.GT.3)) GO TO 590
      N=(N-1)+20
      CALL Select(N,PARAM,Pstr)
      GO TO LAMP
C
C     ---------------------------------------------------------------
C
C         Accessory Mode Selection
C
C     ---------------------------------------------------------------
C
  600 CALL Partable(PARAM)
      DO 610 I=1,2
        Access(I)='OFF'
        ASCII=(I-1)+48
        Pcode=CHAR(ASCII)
        Command=Accoff//Pcode//CSM
        WRITE (UNIT=38,FMT=*,IOSTAT=N,ERR=999) Command
        READ (UNIT=38,FMT=620,IOSTAT=N,ERR=999) Response
        IF (Response(2:2).EQ.'P') THEN
          Access(I)='ON'
          Command=Accon//Pcode//CSM
          CALL Send(Command)
        END IF
  610 CONTINUE
  620 FORMAT (A64)
  630 WRITE (1,*) HOME,CLR
      TITLE='Accessory Control'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,'(T15,A5,T27,A8,T46,A4)') 'INDEX','FUNCTION','MODE'
      CALL Line(NCOL)
      WRITE (1,*)
      WRITE (1,640) '1: ...TEMPERATURE READOUT... ',Access(1)
      WRITE (1,640) '2: ...PRINTER............... ',Access(2)
      IF (Access(2).EQ.'ON') THEN
        WRITE (1,'(T49,A2,A3,A12)') UP,' : ',Printer(IP)
      END IF
      WRITE (1,640) 'X: ...EXIT TO MENU.......... ',' '
      WRITE (1,*)
      CALL Line(NCOL)
  640 FORMAT (T16,A30,A4)
      WRITE (1,*)
  650 WRITE (1,'(T15,A2,A2,A9,A,A)') UP,ERASE,'INDEX #: ',BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'X') GO TO ADVANCED
      I=ICHAR(Code)-48
      IF ((I.LT.1).OR.(I.GT.2)) GO TO 650
      ASCII=(I-1)+48
      Pcode=CHAR(ASCII)
  660 WRITE (1,'(T14,A2,A2,A)') UP,ERASE,'_'
```

```
      WRITE (1,*) '(OFF=0, ON=1) ? ',BELL,'_'
      READ (1,'(I2)',ERR=660) ASCII
      Code=CHAR(ASCII+48)
      IF ((Code.NE.'0').AND.(Code.NE.'1')) GO TO 660
      Command='@F'//Code//Pcode//CSM
      CALL Send(Command)
      IF (Code.EQ.'0') GO TO 699
      IF (I.EQ.1) THEN
        N=9
        CALL Select(N,PARAM,Pstr)
        GO TO 699
      END IF
C     ------------------------------------------------------------
      WRITE (1,*) HOME,CLR
      TITLE='Printer Mode'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,'(T15,A5,T27,A8,T46,A4)') 'INDEX','FUNCTION','MODE'
      CALL Line(NCOL)
      WRITE (1,'(T46,A14)') Pstr(40,PARAM(40))
      WRITE (1,670) '1: .....WAVELENGTH.......... '
      WRITE (1,670) '2: .....TIME................ '
      WRITE (1,670) '3: .....TEMPERATURE......... '
      WRITE (1,*)
      CALL Line(NCOL)
  670 FORMAT (T16,A30)
      WRITE (1,*)
  680 WRITE (1,'(T15,A2,A2,A9,A,A)') UP,ERASE,'INDEX #: ',BELL,'_'
      READ (1,'(I2)',ERR=680) IP
      IF ((IP.LT.1).OR.(IP.GT.3)) GO TO 680
      ASCII=(IP-1)+48
      Pcode=CHAR(ASCII)
  690 WRITE (1,'(T15,A2,A2,A11,A,A)') UP,ERASE,'INTERVAL = ',BELL,'_'
      READ (1,*,ERR=690) NUMBER
      CALL Str(NUMBER,String,4)
      Command='@M'//Pcode//String(1:5)//'!0'
      CALL Send(Command)
  699 Command='@DF0'                 ! Update Cary Accessory Display
      CALL Send(Command)
      GO TO 600
C
```

```
C      ------------------------------------------------------------
C
C           Store Spectrum
C
C      ------------------------------------------------------------
C
  700 IF ((Sstat.NE.'ACQUIRED').AND.(Sstat.NE.'STORED')) THEN
        WRITE (1,*) UP,ERASE,'  Spectrum is ABSENT: ',BELL,'_'
        CALL Wait(2.0)
        WRITE (1,*)
        GO TO 70
      END IF
      IF (ABS(XMIN-ABSC).GT.0.5) THEN
  710   WRITE (1,*) UP,ERASE,'  SCAN ENDED AT',ABSC,' nm (Expected:',
     &  XMIN,'), Proceed (Y or N) ? ',BELL,'_'
        READ (1,80) Icode
        CALL Upper(Icode)
        IF (Icode.EQ.'N') GO TO 70
        IF (Icode.NE.'Y') GO TO 710
      END IF
      TITLE='Store Spectrum'
  720 WRITE (1,*) HOME,CLR
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,*)
      WRITE (1,*) '  Researcher''s Initials, (AA-ZZ) ? ',BELL,'_'
      READ (1,'(A2)') INITIALS
      CALL Upper(INITIALS)
      String='.S'//INITIALS
      WRITE (1,*) DOWN
      WRITE (1,*) UP,ERASE,'  Filename: (16 chars.) ? ',BELL,'_'
      READ (1,'(A16)') Name
      CALL Upper(Name)
      K=16
      DO WHILE (Name(K:K).EQ.' ')
        K=K-1
      END DO
      Fname=Name(1:K)//String(1:4)
      WRITE (1,*) DOWN,'  Directory, (Return = /DEFAULT/) ? ',BELL,'_'
      READ (1,'(A40)') Directory
      IF (Directory.EQ.' ') THEN
        Outfile=Fname
        GO TO 730
      END IF
      L=40
      DO WHILE (Directory(L:L).EQ.' ')
        L=L-1
      END DO
      IF (Directory(L:L).EQ.'/') L=L-1
      Outfile=Directory(1:L)//'/'//Fname
  730 L=63
      DO WHILE (Outfile(L:L).EQ.' ')
        L=L-1
      END DO
  740 WRITE (1,*) DOWN,'  Validating: ',Outfile(1:L),' _'
      OPEN (UNIT=66,FILE=Outfile(1:L),IOSTAT=N,STATUS='NEW')
      WRITE (1,*)
      IF (N.NE.0) THEN
        N=N-500
        WRITE (1,*) UP,ERASE,'_'
        IF (N.EQ.2) WRITE (1,*) ' FILE EXISTS: ',BELL,'_'
        IF (N.EQ.8) WRITE (1,*) ' FILE OPENED: ',BELL,'_'
        IF ((N.NE.2).AND.(N.NE.8)) WRITE (1,*) ' DISK ERROR # ',N,
     &  BELL,' _'
        CLOSE (UNIT=66,STATUS='DELETE')
        CALL Wait(2.0)
        GO TO 720
      END IF
      IF (Code.EQ.'R') GO TO 790
      WRITE (1,*) UP,ERASE,'  Validated Filename: ',Fname
      WRITE (1,*) DOWN,'  Title, (72 chars):'
      WRITE (1,*) '  ',BELL,'_'
      READ (1,750) TITLE
  750 FORMAT (A72)
      WRITE (1,*) DOWN,'  Date, (MM/DD/YY): ',BELL,'_'
      READ (1,760) DATE
  760 FORMAT (A8)
      WRITE (1,*) DOWN
  770 WRITE (1,*) UP,ERASE,'  Concentration, (M): ',BELL,'_'
      READ (1,*,ERR=770) CONC
      IF (CONC.LT.0.0) GO TO 770
      WRITE (1,*) DOWN
  780 WRITE (1,*) UP,ERASE,'  Pathlength, (cm): ',BELL,'_'
      READ (1,*,ERR=780) PATH
      IF (PATH.LT.0.0) GO TO 780
C     ------------------------------------------------------------
  790 J=49     ! # of Parameters
      K=14     ! # of Variables
      VARIABLE(6)=SPECGAIN         ! Store GAIN At Smax
      VARIABLE(10)=SPECBAND        ! Store SBW At Smin
      WRITE  (1,*) DOWN,'  Storing File: ',Outfile(1:L),BELL
      WRITE (66,FMT=750,IOSTAT=N,ERR=820) TITLE
      WRITE (66,FMT=760,IOSTAT=N,ERR=820) DATE
      WRITE (66,FMT=*,IOSTAT=N,ERR=820) XMIN,XMAX,XSTEP,CONC,PATH
      WRITE (66,FMT=*,IOSTAT=N,ERR=820) ORD,ABSC,CELL,CYCLE,SAMPLE,
     &WAVE,TIMER,TEMP,DIST
      WRITE (66,FMT=800,IOSTAT=N,ERR=820) J,K,NARRAY
  800 FORMAT (I3,I3,I6)
      WRITE (66,FMT=810,IOSTAT=N,ERR=820) (PARAM(I),I=1,J)
  810 FORMAT (I2)
      WRITE (66,FMT=*,IOSTAT=N,ERR=820) (VARIABLE(I),I=1,K)
      WRITE (66,FMT=*,IOSTAT=N,ERR=820) (Y(I),I=1,NARRAY)
      WRITE (66,FMT=*,IOSTAT=N,ERR=820) (X(I),I=1,NARRAY)
      CLOSE (UNIT=66,IOSTAT=N,ERR=820,STATUS='KEEP')
      Sstat='STORED'
      CALL Wait(2.0)
      GO TO MENU
C     ------------------------------------------------------------
```

```
  820 WRITE (1,*) UP,ERASE,'  Disk Error #',N,BELL,' :_'
      WRITE (1,*) 'R...RESAVE, X...EXIT to Menu ? ',BELL,'_'
      READ (1,80) Code
      IF ((Code.NE.'R').AND.(Code.NE.'X')) GO TO 820
      WRITE (1,*) UP,ERASE,'  Deleting Old File: ',Outfile(1:L),BELL,
     &' _'
      CLOSE (UNIT=66,IOSTAT=N,ERR=820,STATUS='DELETE')
      CALL Wait(2.0)
      IF (Code.EQ.'R') GO TO 720
      Fname=' '
      Sstat='ACQUIRED'
      GO TO MENU
C
C     ----------------------------------------------------------------
C
C         Exit Program
C
C     ----------------------------------------------------------------
C
  900 IF (Sstat.EQ.'ACQUIRED') THEN
        WRITE (1,*) UP,ERASE,'  SPECTRUM NOT STORED:',
     &  '   Exit (Y or N) ? ',BELL,'_'
        READ (1,80) Code
        CALL Upper(Code)
        IF (Code.EQ.'N') GO TO 70
        IF (Code.NE.'Y') GO TO 900
      END IF
  910 WRITE (1,*) UP,ERASE,'  Set To STANDBY, (Y or N) ? ',BELL,'_'
      READ (1,80) Code
      CALL Upper(Code)
      IF (Code.EQ.'N') GO TO 920
      IF (Code.NE.'Y') GO TO 910
      CALL Send(Standby)
  920 WRITE (1,*) UP,ERASE,UP
      STOP
C
C     ----------------------------------------------------------------
C
C         IEEE-488 Error Exit
C
C     ----------------------------------------------------------------
C
  999 WRITE (1,*) ' Error #',N
      STOP
      END
```

```
C
C     ******************** END OF MAIN PROGRAM **********************
C
C         BLOCK DATA FOR NAMED COMMON BLOCK INITIALIZATION
C
C     ****************************************************************
C
$ALIAS /MODE/,NOALLOCATE
$ALIAS /CARY/,NOALLOCATE
$ALIAS /IP/,NOALLOCATE
$ALIAS /IS/,NOALLOCATE
      BLOCK DATA Arrays
      INTEGER NDATA,XMODE,YMODE
      INTEGER NPAR(49),OFFSET(49)
      REAL ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST
      CHARACTER Pname(49)*10,Vname(14)*8
      COMMON /MODE/NDATA,XMODE,YMODE
      COMMON /CARY/ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST
      COMMON /IP/NPAR,OFFSET
      COMMON /IS/Pname,Vname
C
      DATA NDATA,XMODE,YMODE/0,0,0/
      DATA ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST/0.0,0.0,0.0,
     &0.0,0.0,0.0,0.0,0.0,0.0/
C
      DATA NPAR/6,4,11,11,4,5,9,7,4,5,11,16,8,10,4,2,2,2,2,4,3,3,2,3,
     &3,3,2,6,2,4,2,5,5,2,2,0,0,5,0,2,0,0,0,0,0,0,0,0,0/
C
      DATA OFFSET/0,0,0,4,0,0,0,9,11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     &0,0,0,0,2,6,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/
C
      DATA Pname/'Ordinate','Abscissa','Scan Rate','Chart/cm',
     &'Ref.Mode','Pen Functn','A range','%T Range','Temp.Range',
     &'Log Zero','Der.Range','Der.Temp','Kinet.Rnge','Temp.Zero',
     &'Period','Beam Chge','Sequencer','Auto.Op','Chart Op',
     &'Lamp Pwr','Lamp Sel','Det.Sel','Slit Hgt','BLN Det',
     &'BLN Lamp','BLN Ref','BLN Slit','No.cells','Sample',
     &'Curv.Fit','Rec.Prec','Method','Result','Auto Kin',
     &'Samp/Av.',' ',' ','BLN Stat',' ','Printer',' ',' ',
     &'TEST','TEST','TEST','TEST','TEST','TEST','TEST'/
C
      DATA Vname/'A zero','BL SBW/G','BL Wmax','BL Wmin',
     &'Cyc Time','Sel Gain','Ncycles','Nwlngths','Rec Time',
     &'Sel SBW','%T Zero','SEQ Wmax','SEQ Wmin','Distance'/
C
      END
```

```
C
C      ----------------------------------------------------------------
C
C             Select Mode of Operation for Specific Parameter
C
C      ----------------------------------------------------------------
C
$ALIAS /IP/,NOALLOCATE
$ALIAS /IS/,NOALLOCATE
      SUBROUTINE Select(N,PARAM,Pstr)
      INTEGER ASCII,I,J,K,N,NCOL
      INTEGER NPAR(49),OFFSET(49),PARAM(49),INDEX(11)
      CHARACTER Pname(49)*10,Pstr(49,16)*14,Vname(14)*8
      CHARACTER Command*44,CSM,Icode,Key*2,Pcode,Parset*2,TITLE*72
      CHARACTER BELL,CLR*2,ESC,ERASE*2,DOWN*2,HOME*2,UP*2
      COMMON /IP/NPAR,OFFSET,/IS/Pname,Vname
      DATA CSM,Key,Parset/'0','@D','@H'/
      DATA (INDEX(I),I=1,11)/1,3,4,6,7,9,10,12,13,15,16/
      BELL=CHAR(7)
      ESC=CHAR(27)
      CLR=ESC//'J'
      DOWN=ESC//'B'
      ERASE=ESC//'K'
      HOME=ESC//'h'
      UP=ESC//'A'
      NCOL=50
   10 WRITE (1,*) HOME,CLR
      TITLE='Operating Mode Selection'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,'(T15,A5,T30,A10)') 'Index',Pname(N)
      CALL Line(NCOL)
      WRITE (1,*)
      DO 20 I=1,NPAR(N)
        J=I+OFFSET(N)
        IF ((N.EQ.4).AND.(J.EQ.5)) J=1            ! Chart Index Offset
        IF (N.EQ.11) THEN                         ! Derivative Modes
          J=INDEX(I)                              ! Use Valid Index
          IF ((PARAM(1).NE.0).AND.(I.LE.4)) GO TO 20
        END IF
        WRITE (1,30) I,': ........',Pstr(N,J)
   20 CONTINUE
   30 FORMAT (T16,I2,A10,T30,A14)
      WRITE (1,*)
      CALL Line(NCOL)
      WRITE (1,*)
   40 WRITE (1,'(T15,A2,A2,A9,A,A)') UP,ERASE,'Index #: ',BELL,'_'
      READ (1,'(I2)',ERR=40) K
      IF ((K.LT.1).OR.(K.GT.NPAR(N))) GO TO 40    ! Invalid Index Entry
      IF (PARAM(1).NE.0) THEN
        IF ((K.EQ.5).AND.(N.EQ.6)) GO TO 40       ! Only Log(Abs) Valid
        IF ((K.LE.4).AND.(N.EQ.11)) GO TO 40      ! Invalid Deriv Index
      END IF
      IF ((N.EQ.38).AND.(K.GT.2)) GO TO 60        ! Baseline Setup Mode
      IF (N.EQ.11) K=INDEX(K)                         ! Index To Deriv Mode
   50 ASCII=(K-1)+OFFSET(N)+48
      IF ((N.EQ.4).AND.(K.EQ.1)) ASCII=ASCII-4        ! Chart Index Offset
      Icode=CHAR(ASCII)
      ASCII=(N-1)+48
      Pcode=CHAR(ASCII)
      Command=Parset//Pcode//Icode//CSM
      CALL Send(Command)
      IF ((N.EQ.6).AND.(K.GT.2)) THEN                 ! Special Pen Modes
        N=11                                          ! Derivative Modes
        IF (K.EQ.5) N=10                              ! Log(Abs) Mode
        GO TO 10                                      ! Select Setting
      END IF
      N=K
      RETURN
   60 IF (K.EQ.4) GO TO 40
      Pcode=CHAR(48+N-1)
      Command=Parset//Pcode//'0'//CSM             ! Set Status To OFF
      CALL Send(Command)
      Command=Key//'10'                           ! Key = 1
      CALL Send(Command)
      Command=Key//'=0'                           ! Key = ENTER
      CALL Send(Command)
      Command=Key//'h0'                           ! Key = RIGHT CURSOR
      IF (K.EQ.3) GO TO 70
      CALL Send(Command)
   70 CALL Send(Command)
      CALL Send(Command)
      Command=Key//'=0'                           ! Key = ENTER
      CALL Send(Command)
      N=K
      IF (K.EQ.5) N=2
      RETURN
      END
```

```
C
C     ------------------------------------------------------------------
C
C         Baseline Scan Control
C
C     ------------------------------------------------------------------
C
      SUBROUTINE Bline(WMIN,WMAX,Bdet,Bgain,Blamp,Bperiod,Brate,Bref,
     &Bsbw,Bslit,Bscan,Btime,MATCH,MODE)
      INTEGER INDEX,MODE,N,NCOL
      REAL NUMBER,WMIN,WMAX
      LOGICAL MATCH,MONITOR
      CHARACTER*(*) Bdet,Bgain,Blamp,Bperiod,Brate,Bref
      CHARACTER*(*) Bsbw,Bslit,Bscan,Btime
      CHARACTER Bgbw*4,BELL,Code,CLR*2,DOWN*2,ERASE*2,Esc,HOME*2,UP*2
      CHARACTER Bmin*6,Bmax*6,Command*44,Response*64,String*14
      CHARACTER Autobal*4,Blstat*5,Start*4,TITLE*72
      DATA Autobal,Blstat,Start/'@DU0','@G1U0','@DP0'/
      Esc=CHAR(27)
      BELL=CHAR(7)
      CLR=Esc//'J'
      DOWN=Esc//'B'
      ERASE=Esc//'K'
      HOME=Esc//'h'
      UP=Esc//'A'
      MONITOR=.FALSE.                      ! For Testing Routine ONLY
      NCOL=70
      IF (WMAX.GT.800.0) WMAX=WMAX+0.2     ! * Cary Baseline Bug Fix *
      CALL Str(WMAX,String,5)
      Bmax=String(2:7)
      IF (WMIN.GT.800.0) WMIN=WMIN-0.2     ! * Cary Baseline Bug Fix *
      CALL Str(WMIN,String,5)
      Bmin=String(2:7)
      Bgbw=Bsbw                            ! Only One Of SBW Or GAIN Is
      IF (MODE.EQ.2) Bgbw=Bgain            ! Stored By Cary For Baseline
C     ------------------------------------------------------------------
      WRITE (1,*) HOME,CLR
      TITLE='Baseline Scan Control'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,10) DOWN,'  Wavelength Limits, (nm): ',WMAX,' / ',WMIN
   10 FORMAT (T2,A2,A27,F4.1,A3,F4.1)
      WRITE (1,*) DOWN,'  Scan Rate, (nm/sec)     : ',Bscan
      WRITE (1,*) DOWN,'  Response Time, (sec)    : ',Btime
      IF (MODE.EQ.1) THEN
        WRITE (1,*) DOWN,'  Spectral Bandwidth,(nm): ',Bsbw
        GO TO 20
      END IF
      WRITE (1,*) DOWN,'  AUTOSLIT Gain Level     : ',Bgain
   20 WRITE (1,*) DOWN
      WRITE (1,*) DOWN,'  Place Solvent Cells In BOTH Beams:'
      WRITE (1,*) DOWN,'       S.....Start Scan'
      WRITE (1,*) DOWN,'       A.....Abort Scan'
      WRITE (1,*) DOWN,'  Enter the CODE. ',BELL,'_'
   30 READ (1,'(A1)') Code
      CALL Upper(Code)
      IF (Code.EQ.'A') THEN
        MATCH-.FALSE.
        RETURN
      END IF
      IF (Code.NE.'S') GO TO 30
      WRITE (1,*) UP,ERASE,UP,UP,ERASE,UP,UP,ERASE,UP,UP,ERASE,'_'
      WRITE (1,*) ' Sending Baseline Parameters: ',BELL
      Command-'@J'//Bmax//'!'//Bmin//'!'//Bgbw//'!'//Bref//'!'//
     &Blamp//'!'//Bdet//'!'//Bslit//'!'//Brate//'!'//Bperiod//'!0'
      WRITE (UNIT-38,FMT-*,IOSTAT-N,ERR-999) Command
      READ (UNIT-38,FMT-40,IOSTAT-N,ERR-999) Response
   40 FORMAT (A64)
      IF (Response(2:2).EQ.'N') THEN
        INDEX-ICHAR(Response(4:4))-48
        WRITE (1,*) UP,ERASE,'  Parameter Error: ',INDEX,BELL
        CALL Wait(2.0)
        MATCH-.FALSE.
        RETURN
      END IF
      WRITE (1,*) UP,ERASE,'  Recording Baseline:',BELL
      CALL Send(Start)
   50 Command-Blstat
      IF (MONITOR) WRITE (1,*) ' Command - ',Command
      WRITE (UNIT-38,FMT-*,IOSTAT-N,ERR-999) Command
      READ (UNIT-38,FMT-40,IOSTAT-N,ERR-999) Response
      IF (MONITOR) WRITE (1,*) ' Response - ',Response
      INDEX-ICHAR(Response(6:6))-48
      IF (INDEX.NE.1) GO TO 50
      WRITE (1,*) UP,ERASE,'  Performing Auto Balance:',BELL
      CALL Send(Autobal)
      CALL Wait(2.0)
      RETURN
  999 WRITE (1,*) ' Error #',N,' in SUBROUTINE Bline'
      STOP
      END
```

```
C
C     ------------------------------------------------------------
C
C         Go To Specified Wavelength
C
C     ------------------------------------------------------------
C
      SUBROUTINE GOTO(Wlength)
      INTEGER LENSTR
      CHARACTER Ascii,CSM,Slew,Model,Ncell,Range,Windex
      CHARACTER Command*4,Key*2
      CHARACTER*(*) Wlength
      CSM='0'
      Key='@D'
      Command=Key//'J'//CSM              ! Key = GOTO WAVELENGTH
      CALL Send(Command)
      LENSTR=LEN(Wlength)
      DO 10 I=1,LENSTR
        Ascii=Wlength(I:I)
        IF (Ascii.EQ.' ') GO TO 10
        IF (Ascii.EQ.'.') Ascii=':'
        Command=Key//Ascii//CSM          ! Key = NUMBER (0-9)
        CALL Send(Command)
   10 CONTINUE
      Command=Key//'='//CSM              ! Key = ENTER
      CALL Send(Command)
   20 CALL Instats(Slew,Model,Ncell,Range,Windex)
      IF (Slew.NE.'0') GO TO 20
      RETURN
      END
C
C     ------------------------------------------------------------
C
C         Instrument Status Test
C
C     ------------------------------------------------------------
C
      SUBROUTINE Instats(Slew,Model,Ncell,Range,Windex)
      INTEGER N
      CHARACTER Slew,Model,Ncell,Range,Windex
      CHARACTER Stats*3,Data*12
      Stats='@B0'
   10 WRITE (UNIT=38,FMT=*,IOSTAT=N,ERR=999) Stats
      READ (UNIT=38,FMT=20,IOSTAT=N,ERR=999) Data
   20 FORMAT (A12)
      Slew=Data(4:4)
      Model=Data(5:5)
      Ncell=Data(6:6)
      Range=Data(7:7)
      Windex=Data(8:8)
      RETURN
  999 WRITE (1,*) ' Error #',N,' in SUBROUTINE Instats'
      STOP
      END
```

```
C
C     ----------------------------------------------------------------
C
C         Read Parameter Table From CARY 2390
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Partable(PARAM)
      INTEGER LENSTR,N,INDEX,PARAM(49)
      LOGICAL TEST
      CHARACTER Command*3,Response*64,Ascii
      TEST-.FALSE.
      Command-'@E0'
      WRITE (UNIT-38,FMT-*,IOSTAT-N,ERR-999) Command
      READ (UNIT-38,FMT-10,IOSTAT-N,ERR-999) Response
   10 FORMAT (A64)
      IF (TEST) WRITE (1,*) ' RESPONSE - ',Response
      Ascii-Response(4:4)
      LENSTR-ICHAR(Ascii)-48
      IF (TEST) WRITE (1,*) ' String Length   -',LENSTR
      DO 20 I-1,LENSTR
        J-I+4
        Ascii-Response(J:J)
        IF (TEST) WRITE (1,*) ' ASCII Character - ',Ascii
        INDEX-ICHAR(Ascii)-48
        PARAM(I)-INDEX
        IF (TEST) WRITE (1,*) ' Parameter Index -',PARAM(I)
   20 CONTINUE
      RETURN
  999 WRITE (1,*) ' Error #',N,' in SUBROUTINE Partable'
      STOP
      END
C
C     ----------------------------------------------------------------
C
C         Read Variable Table From CARY 2390
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Vartable(VARIABLE)
      INTEGER LENSTR(14),N
      REAL NUMBER,VARIABLE(14)
      LOGICAL TEST
      CHARACTER Ascii,CSM,Command*5,Response*64,String*14,Varout*3
      DATA (LENSTR(I),I-1,14)/14,11,11,11,10,10,8,8,8,11,11,11,11,11/
      TEST-.FALSE.
      CSM-'0'
      Varout-'@C2'
      DO 10 I-1,14
        J-I-1
        Ascii-CHAR(J+48)
        Command-Varout//Ascii//CSM
        IF (TEST) WRITE (1,*) ' Command  - ',Command
        WRITE (UNIT-38,FMT-*,IOSTAT=N,ERR=999) Command
        READ (UNIT=38,FMT=20,IOSTAT=N,ERR=999) Response
        IF (TEST) WRITE (1,*) ' Response - ',Response
        String=Response(6:6+LENSTR(I))
        IF (TEST) WRITE (1,*) ' String = ',String
        CALL Val(String,NUMBER)
        VARIABLE(I)=NUMBER
        IF (TEST) WRITE (1,*) ' VALUE  =',VARIABLE(I)
   10 CONTINUE
   20 FORMAT (A64)
      RETURN
  999 WRITE (1,*) ' Error #',N,' in SUBROUTINE Vartable'
      STOP
      END
C
C     ----------------------------------------------------------------
C
C         Print a TITLE Centered in 72 columns
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Center(TITLE)
      INTEGER I,J,N
      CHARACTER TITLE*72,BLANK*36
      BLANK='                                    '
      I=72
      J=0
      DO WHILE (ICHAR(TITLE(I:I)).EQ.32)
        J=J+1
        I=72-J
      END DO
      N=J/2
      WRITE (1,*) BLANK(1:N),TITLE(1:I)
      RETURN
      END
C
C     ----------------------------------------------------------------
C
C         Print a line of N '-' characters (72 columns max)
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Line(N)
      INTEGER I,N
      CHARACTER BLANK*72,DLINE*72,SPACE*36
      SPACE='                                    '
      BLANK=SPACE//SPACE
      SPACE='------------------------------------'
      DLINE=SPACE//SPACE
      IF (N.GT.72) N=72
      I=(72-N)/2
      WRITE (1,*) BLANK(1:I),DLINE(1:N)
      RETURN
      END
```

```
C
C     ----------------------------------------------------------------
C
C         Enter and Validate Wavelength Limits
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Limits(MIN,MAX)
      REAL MIN,MAX,SWAP
      CHARACTER BELL,ERASE*2,ESC,UP*2
      BELL=CHAR(7)
      ESC=CHAR(27)
      ERASE=ESC//'K'
      UP=ESC//'A'
   10 WRITE (1,*) UP,ERASE,'_'
      WRITE (1,*) ' Wavelength Limits: (Min,Max) ',ERASE,BELL,'_'
      READ (1,*,ERR=10) MIN,MAX
      MIN=ABS(MIN)
      MAX=ABS(MAX)
      IF (MIN.LT.MAX) GO TO 20
      SWAP=MIN
      MIN=MAX
      MAX=SWAP
   20 MIN=INT(MIN+.5)
      MAX=INT(MAX+.5)
      IF (MIN.LT.185) GO TO 10
      IF (MAX.GT.3152) GO TO 10
      RETURN
      END
C
C     ----------------------------------------------------------------
C
C         Convert String Entry To Uppercase If Required
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Upper(Code)
      INTEGER LENSTR,N
      CHARACTER*(*) Code
      LENSTR=LEN(Code)
      DO 10 I=1,LENSTR
        N=ICHAR(Code(I:I))
        IF (N.GT.96) Code(I:I)=CHAR(N-32)
   10 CONTINUE
      RETURN
      END
```

```
C
C     ----------------------------------------------------------------
C
C         Read Data In Real Time (INTERVAL) Mode From CARY 2390
C
C     ----------------------------------------------------------------
C
$EMA/DATA/
      SUBROUTINE Acquire(Inc,PRINT,SINGLE,WAVELENGTH)
      INTEGER N,NCOL,NDATA,XMODE,YMODE,XOFF(4),YOFF(6)
      REAL ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST
      REAL X(10001),Y(10001),WAVELENGTH
      CHARACTER Command*10,Data*64,Inc*4,Esc,DOWN*2,ERASE*2,UP*2
      CHARACTER S1*8,S2*8,S3,S4,S5*3,S6*8,S7*6,S8*6,S9*7
      LOGICAL CHECK,PRINT,SINGLE,TEST
      COMMON /MODE/NDATA,XMODE,YMODE
      COMMON /CARY/ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST
      COMMON /DATA/Y,X
      DATA (XOFF(I),I-1,4)/7,5,5,6/
      DATA (YOFF(I),I-1,6)/7,6,5,6,11,6/
      Esc-CHAR(27)
      UP-Esc//'A'
      DOWN-Esc//'B'
      ERASE-Esc//'K'
      CHECK-.FALSE.                ! Only Used For Testing Routine
      TEST-.FALSE.                 ! Only Used For Testing Routine
      J-2+YOFF(YMODE)              ! The First Two Fields In Data String
      K-J+2                        ! Vary In Length With Choice Of Abscissa
      L-K+XOFF(XMODE)              ! And Ordinate - XMODE & YMODE Select
      M-L+2                        ! The Correct Offsets From XOFF/YOFF
      NCOL-70
      IF (.NOT.PRINT) GO TO 20
      CALL Line(NCOL)
      WRITE (1,10) 'Ordinate','Abscissa','Cell','Cycle','Sample',
     &'Wlength','Time','Temp,C','Dist'
   10 FORMAT (A10,A10,A5,A6,A7,A10,A8,A8,A8)
      CALL Line(NCOL)
      WRITE (1,*) DOWN
   20 Command-'@K11'//Inc//'!0'
      IF (TEST) WRITE (1,*) ' Command  - ',Command
      WRITE (UNIT-38,FMT-*,IOSTAT-N,ERR-999) Command
      IF (SINGLE) THEN
        READ (UNIT-38,FMT-30,IOSTAT-N,ERR-999) Data
        S6-Data(M+9:M+16)
        CALL Val(S6,WAVELENGTH)
        RETURN
      END IF
   30 FORMAT (A64)
      DO 100 I-1,NDATA
        READ (UNIT-38,FMT-30,IOSTAT-N,ERR=999) Data
C       IF (CHECK) WRITE (1,30) Data
        S1-Data(2:J)               ! Ordinate - Variable Length Field
        S2-Data(K:L)               ! Abscissa - Variable Length Field
        S3-Data(M:M)               ! Remaining Fields Are Fixed Length
```

```
          S4=Data(M+2:M+3)
          S5=Data(M+5:M+7)
          S6=Data(M+9:M+16)
          S7=Data(M+18:M+23)
          S8=Data(M+25:M+30)
          S9=Data(M+32:M+38)
C         IF (CHECK) WRITE (1,*) S1,S2,S3,S4,S5,S6,S7,S8,S9
          CALL Val(S1,ORD)
          CALL Val(S2,ABSC)
          CALL Val(S3,CELL)
          CALL Val(S4,CYCLE)
          CALL Val(S5,SAMPLE)
          CALL Val(S6,WAVE)
          CALL Val(S7,TIMER)
          CALL Val(S8,TEMP)
          CALL Val(S9,DIST)
          Y(I)=ORD                ! Ordinate And Abscissa Stored In Arrays
          X(I)=ABSC               ! /CARY/ Variables Return Final Reading
        IF (.NOT.PRINT) GO TO 100
        WRITE (1,*) UP,ERASE,UP
        WRITE (1,40) ORD,ABSC,CELL,CYCLE,SAMPLE,WAVE,TIMER,TEMP,DIST
   40   FORMAT (F10.4,F10.2,F5.1,F6.1,F7.1,F10.2,F8.1,F8.2,F8.2)
  100   CONTINUE
        RETURN
  999   WRITE (1,*) 'Error #',N,' in SUBROUTINE Acquire'
        RETURN
        END
```

```
C
C     ---------------------------------------------------------------
C
C         Convert ASCII String To Numeric Value (10 Digits Max'm)
C
C     ---------------------------------------------------------------
C
      SUBROUTINE Val(String,VALUE)
      INTEGER DECPT,EXPON,LENSTR,N,NUM(10)
      REAL VALUE
      DOUBLE PRECISION MULT,SIGN,TEN,DECIMAL
      CHARACTER Ascii
      CHARACTER*(*) String
      LOGICAL INTEGER,TEST
      INTEGER-.TRUE.
      TEST-.FALSE.            ! Only Used For Testing The Routine
      J-1
      K-0
      DECPT-0
      SIGN-1.0
      TEN-10.0
      DECIMAL-0.0
      LENSTR-LEN(String)
      IF (TEST) WRITE (1,*) ' String Number - ',String
      IF (TEST) WRITE (1,*) ' String Length -',LENSTR
      DO 100 I-1,LENSTR
        Ascii-String(I:I)
        N-ICHAR(Ascii)
        IF ((N.GE.48).AND.(N.LE.57)) GO TO 20
        IF (N.EQ.46) INTEGER-.FALSE.
        IF (N.EQ.46) DECPT-K
        IF (N.EQ.45) SIGN--1.0
        GO TO 100
   20 NUM(J)-N-48
      K-J
      J-J+1
  100 CONTINUE
      IF ((DECPT.EQ.0).AND.(INTEGER)) DECPT-K
      DO 200 J-1,K
        EXPON-DECPT-J
        MULT-TEN**EXPON
        DECIMAL-DECIMAL+NUM(J)*MULT
  200 CONTINUE
      VALUE-SIGN*DECIMAL
      IF (TEST) WRITE (1,*) ' Value -',VALUE
      RETURN
      END
```

```
C
C     ----------------------------------------------------------------
C
C         Convert Number To ASCII String
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Str(VALUE,String,PREC)
      INTEGER ASCII,DECPT,I,J,LENSTR,NDIGIT,NUMBER,PREC
      REAL VALUE
      DOUBLE PRECISION DECIMAL,FRACTION,TEN
      CHARACTER Concat*14,Digit(12),Sign,String*14
      LOGICAL INTEGER,TEST
      INTEGER=.TRUE.
      TEST=.FALSE.           ! Only Used For Testing The Routine
      DECPT=0
      J=0
      TEN=10.0
      Sign=' '
      Concat=' '
      IF (TEST) WRITE (1,*) ' Value Entered = ',VALUE
      IF (VALUE.LT.0.0) Sign='-'
      IF (VALUE.EQ.0.0) GO TO 100
      DECIMAL=ABS(VALUE)
      DO WHILE (DECIMAL.GE.1.0)
        DECIMAL=DECIMAL/TEN
        J=J+1
      END DO
      DECPT=J
      IF (TEST) WRITE (1,*) ' # of Whole Digits: ',DECPT
      IF (DECPT.EQ.0) GO TO 30
      DO 20 J=1,DECPT
        DECIMAL=DECIMAL*TEN
        NUMBER=INT(DECIMAL)
        ASCII=NUMBER+48
        Digit(J)=CHAR(ASCII)
        FRACTION=DECIMAL-NUMBER
        DECIMAL=DINT(FRACTION*TEN**(PREC-J)+.5)/TEN**(PREC-J)
   20 CONTINUE
      IF (.NOT.TEST) GO TO 30
      WRITE (1,*) ' The Whole Digits = ',(Digit(I), I=1,DECPT)
   30 J=DECPT
      IF (TEST) WRITE (1,*) ' Decimal Fraction = ',DECIMAL
      IF (DECIMAL.NE.0.0) INTEGER=.FALSE.
      IF (DECPT.GE.12) GO TO 40
      DO WHILE (DECIMAL.NE.0.0)
        J=J+1
        DECIMAL=DECIMAL*TEN
        NUMBER=INT(DECIMAL)
        ASCII=NUMBER+48
        Digit(J)=CHAR(ASCII)
        FRACTION=DECIMAL-NUMBER
        DECIMAL=DINT(FRACTION*TEN**(PREC-J)+.5)/TEN**(PREC-J)
```

```
          IF (DECIMAL.EQ.1.0) THEN
            DIGIT(J)-CHAR(ASCII+1)
            DECIMAL-0.0
          END IF
          IF (J.GE.12) DECIMAL-0.0
        END DO
 40     NDIGIT-J
        IF (.NOT.TEST) GO TO 50
        WRITE (1,*) ' The Characters  - ',(Digit(I), I-1,NDIGIT)
 50     IF (NDIGIT.GT.12) GO TO 200
        DO 60 I-1,NDIGIT
          Concat(I:I)-Digit(I)
 60     CONTINUE
        IF (INTEGER) GO TO 80
        IF (DECPT.EQ.0) GO TO 70
        String-Sign//Concat(1:DECPT)//'.'//Concat(DECPT+1:14)
        RETURN
 70     String-Sign//'.'//Concat
        RETURN
 80     String-Sign//Concat
        RETURN
100     String-' 0.0'
        RETURN
200     WRITE (1,*) ' Error in data: (too many digits)'
        STOP
        END
```

```
C
C     ----------------------------------------------------------------
C
C         Send a Command String To CARY 2390
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Send(Command)
      INTEGER N
      CHARACTER*(*) Command
      CHARACTER Response*64
      LOGICAL TEST
      TEST=.FALSE.              ! Only Used For Testing The Routine
      IF (TEST) WRITE (1,*) ' Command  - ',Command
      WRITE(UNIT=38,FMT=*,IOSTAT=N,ERR=999) Command
      READ (UNIT=38,FMT=10,IOSTAT=N,ERR=999) Response
   10 FORMAT (A64)
      IF (TEST) WRITE (1,*) ' Response - ',Response
      RETURN
  999 WRITE (1,*) 'Error #',N,' in SUBROUTINE Send'
      RETURN
      END
C
C     ----------------------------------------------------------------
C
C         TERMINATE Real Time Transmission from CARY 2390
C
C     ----------------------------------------------------------------
C
C     Send UNTALK/UNLISTEN = '_?' to IEEE-488 Bus
C
      SUBROUTINE Terminate
C     CALL CMDW(35,'_?',0)  ! CMDW occasionally fails to UNADDRESS
      CALL ABRT(35,3)       ! The ABRT command sends '_?' reliably
      RETURN
      END
C
C     ----------------------------------------------------------------
C
C         Wait Specified Delay (sec)
C
C     ----------------------------------------------------------------
C
      SUBROUTINE Wait(DELAY)
      REAL DELAY,PERIOD,Tzero,Time
      PERIOD=0.0
      Tzero=Time(I)
      DO WHILE (PERIOD.LT.DELAY)
        PERIOD=Time(I)-Tzero
      END DO
      RETURN
      END
```

```
C
C     ---------------------------------------------------------------
C
C         Read Time (sec) from the HP 1000's RTE-6 Operating System
C
C         Note: I is a dummy argument, no values are passed
C
C     ---------------------------------------------------------------
C
      REAL FUNCTION Time(I)
      INTEGER ICODE,ITIME(5)
      ICODE-11
      CALL EXEC(ICODE,ITIME)
      Time-FLOAT(ITIME(1))/100.0+FLOAT(ITIME(2))+FLOAT(ITIME(3))*60.0
     &+FLOAT(ITIME(4))*3600.0
      RETURN
      END
```